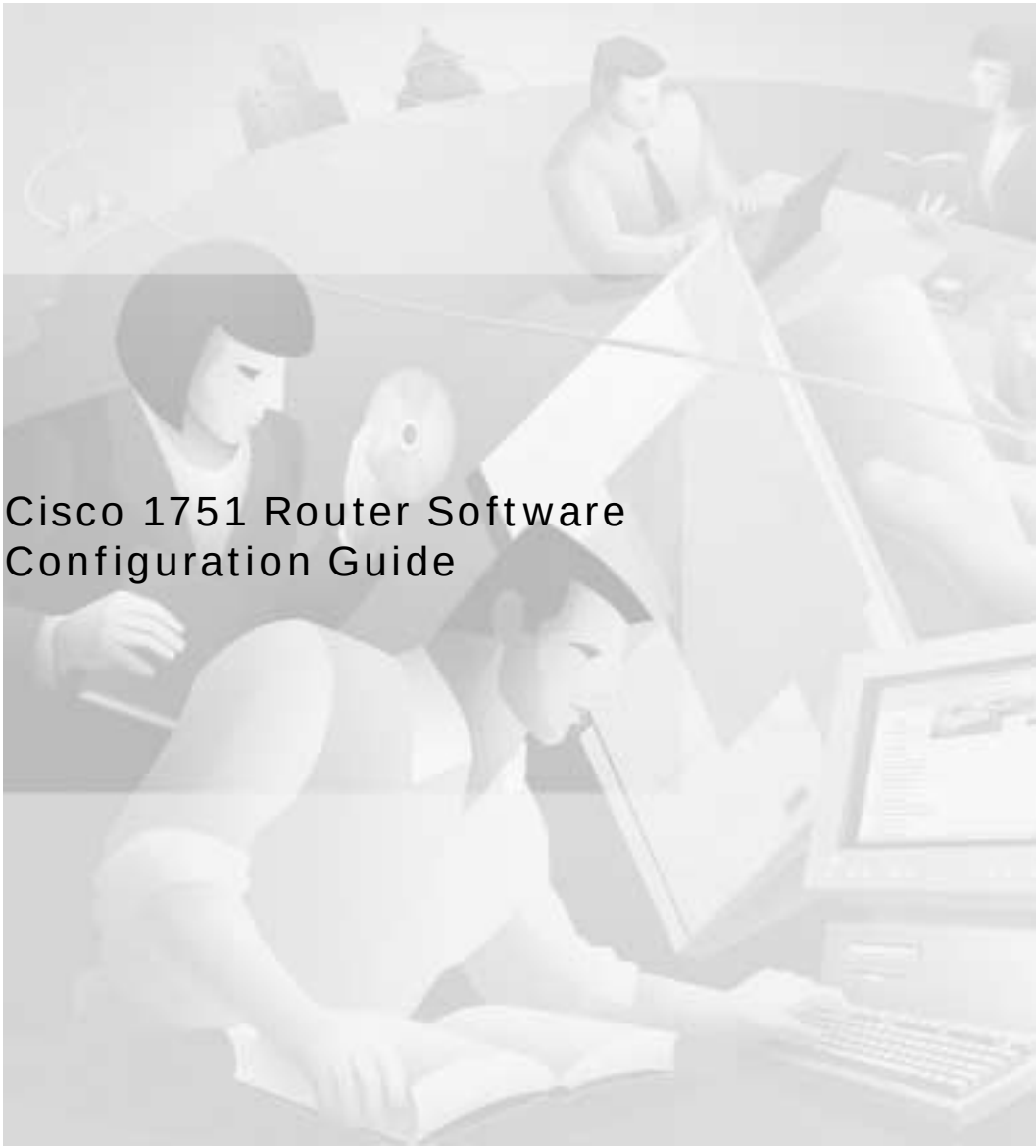


# Cisco 1751 Router Software Configuration Guide

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number:
Text Part Number: OL-1070-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AtmDirector, Browse with Me, CCDA, CCDE, CCDP, CCIE, CCNA, CCNP, CCSI, CD-PAC, *CiscoLink*, the Cisco Net*Works* logo, the Cisco Powered Network logo, Cisco Systems Networking Academy, the Cisco Systems Networking Academy logo, Fast Step, Follow Me Browsing, FormShare,FrameShare, GigaStack, IGX, Internet Quotient, IP/VC, iQ Breakthrough, iQ Expertise, iQ FastTrack, the iQ Logo, iQ Net Readiness Scorecard, MGX, the Networkers logo, *Packet*, PIX, RateMUX, ScriptShare, SlideCast, SMARTnet, TransPath, Voice LAN, Wavelength Router, WebViewer are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, Empowering the Internet Generation, are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, Cisco, the Cisco Certified Internetwork Expert Logo, Cisco IOS, the Cisco IOS logo, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Enterprise/Solver, EtherChannel, EtherSwitch, FastHub, IOS, IP/TV, LightStream, Post-Routing, Pre-Routing, Registrar, StrataView Plus, Stratm, SwitchProbe, TeleRouter, and VCO are registered trademarks of Cisco Systems, Inc. or its affiliates in the U.S. and certain other countries.

All other brands, names, or trademarks mentioned in this document or Web site are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0011R)

*Cisco 1751 Router Software Configuration Guide*
Copyright © 2001, Cisco Systems, Inc.
All rights reserved.

# CONTENTS


**About This Guide xi**

Objectives **xi**

Audience **xi**

Cisco IOS Software Documentation **xi**

Organization **xiv**

Command Syntax Conventions **xiv**

Cisco Connection Online **xv**

Documentation Feedback **xv**

CHAPTER 1 **Voice over IP Overview 1-1**

Voice Primer **1-1**

How VoIP Processes a Typical Telephone Call **1-2**

Numbering Scheme **1-2**

Analog Compared with Digital **1-3**

CODECs **1-3**

Mean Opinion Score **1-3**

Delay **1-4**

Jitter **1-5**

End-to-End Delay **1-5**

Echo **1-5**

Signaling **1-6**

CHAPTER 2 **VoIP Configuration 2-1**

Prerequisite Tasks **2-1**

Configuration Tasks **2-2**

Configure IP Networks for Real-Time Voice Traffic **2-2**

Configure RSVP for Voice **2-3**

Enable RSVP **2-3**

RSVP Configuration Example **2-4**

Configure Multilink PPP with Interleaving **2-4**

Multilink PPP Configuration Example **2-5**

Configure RTP Header Compression **2-6**

Enable RTP Header Compression on a Serial Interface **2-7**

Change the Number of Header Compression Connections **2-7**

RTP Header Compression Configuration Example **2-7**

Configure Custom Queuing **2-7**

Configure Weighted Fair Queuing **2-7**

Configure Number Expansion **2-8**

Create a Number Expansion Table **2-8**

Configure Number Expansion **2-9**

Configure Dial Peers **2-9**

Inbound versus Outbound Dial Peers **2-10**

Create a Dial-Peer Configuration Table **2-12**

Configure POTS Dial Peers **2-12**

Outbound Dialing on POTS Dial Peers **2-13**

Configure VoIP Dial Peers **2-13**

Verifying Your Configuration **2-14**

Troubleshooting Tips **2-14**

Configure Voice Ports **2-14**

Configure FXS or FXO Voice Ports **2-15**
Verifying Your Configuration **2-16**
Troubleshooting Tips **2-16**
Fine-Tune FXS and FXO Voice Ports **2-16**
Configure E&M Voice Ports **2-18**
Verifying Your Configuration **2-19**
Troubleshooting Tips **2-20**
Fine-Tune E&M Voice Ports **2-20**
Additional VoIP Dial Peer Configurations **2-21**
Configure IP Precedence for Dial Peers **2-22**
Configure RSVP for Dial Peers **2-22**
Configure CODEC and VAD for Dial Peers **2-23**
Configure CODEC for a VoIP Dial Peer **2-23**
Configure VAD for a VoIP Dial Peer **2-24**
Configure Frame Relay for VoIP **2-24**
Frame Relay for VoIP Configuration Example **2-25**
Configure Microsoft NetMeeting for VoIP **2-26**
Configure VoIP to Support Microsoft NetMeeting **2-26**
Configure Microsoft NetMeeting for VoIP **2-26**
Initiate a Call Using Microsoft NetMeeting **2-27**

CHAPTER 3 **VoIP Configuration Examples 3-1**

FXS-to-FXS Connection Using RSVP **3-1**
Configuration for Router RLB-1 **3-2**
Configuration for Router RLB-w **3-3**
Configuration for Router RLB-e **3-4**
Configuration for Router RLB-2 **3-5**

Linking PBX Users with E&M Trunk Lines **3-5**
Router SJ Configuration **3-6**
Router SLC Configuration **3-7**
FXO Gateway to PSTN **3-7**
Router SJ Configuration **3-8**
Router SLC Configuration **3-8**
FXO Gateway to PSTN (PLAR Mode) **3-9**
Router SJ Configuration **3-9**
Router SLC Configuration **3-10**

CHAPTER 4 **VoIP Commands 4-1**
acc-qos **4-4**
answer-address **4-5**
codec **4-6**
comfort-noise **4-7**
connection **4-8**
cptone **4-10**
description **4-11**
destination-pattern **4-12**
dial-control-mib **4-13**
dial-peer voice **4-13**
dial-type **4-14**
echo-cancel coverage **4-15**
echo-cancel enable **4-16**
expect-factor **4-17**
fax-rate **4-18**
icpif **4-19**

impedance **4-20**
input gain **4-21**
ip precedence **4-22**
ip udp checksum **4-22**
music-threshold **4-23**
non-linear **4-24**
num-exp **4-25**
operation **4-25**
output attenuation **4-26**
port **4-27**
prefix **4-28**
req-qos **4-29**
ring frequency **4-30**
ring number **4-31**
session protocol **4-32**
session target **4-32**
show call active voice **4-34**
show call history voice **4-37**
show controllers voice **4-40**
show diag **4-42**
show dial-peer voice **4-45**
show dialplan incall number **4-47**
show dialplan number **4-48**
show num-exp **4-48**
show voice dsp **4-49**
show voice port **4-50**
shutdown (dial-peer configuration) **4-55**

shutdown (voice-port configuration) **4-56**
signal **4-56**
snmp enable peer-trap poor-qov **4-58**
snmp-server enable traps **4-59**
snmp trap link-status **4-60**
timeouts initial **4-61**
timeouts interdigit **4-62**
timing **4-63**
type **4-65**
vad **4-67**
voice-port **4-67**

CHAPTER 5 **VoIP Debug Commands 5-1**
Using Debug Commands **5-1**
debug voip ccapi error **5-2**
debug voip ccapi inout **5-2**
debug vpm all **5-5**
debug vpm dsp **5-5**
debug vpm error **5-6**
debug vpm port **5-6**
debug vpm signal **5-7**
debug vpm spi **5-8**
debug vtsp all **5-10**
debug vtsp dsp **5-11**
debug vtsp error **5-11**
debug vtsp port **5-13**
debug vtsp session **5-16**

debug vtsp stats **5-19**

debug vtsp tone **5-20**

debug vtsp vofr subframe **5-20**

CHAPTER 6 **Routing Between Virtual LANs Overview 6-1**

What Is a VLAN? **6-1**

LAN Segmentation **6-2**

Security **6-2**

Broadcast Control **6-3**

Performance **6-3**

Network Management **6-3**

Communication Between VLANs **6-3**

VLAN Colors **6-3**

Why Implement VLANs? **6-4**

Communicating Between VLANs **6-4**

VLAN Translation **6-4**

Designing Switched VLANs **6-4**

CHAPTER 7 **Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation 7-1**

IEEE 802.1Q Encapsulation Configuration Task List **7-1**

Configuring AppleTalk Routing over IEEE 802.1Q **7-1**

Enabling AppleTalk Routing **7-2**

Configuring AppleTalk on the Subinterface **7-2**

Defining the VLAN Encapsulation Format **7-2**

Configuring IP Routing over IEEE 802.1Q **7-3**

Enabling IP Routing **7-3**

Defining the VLAN Encapsulation Format **7-3**

Assigning IP Address to Network Interface **7-3**
Configuring IPX Routing over IEEE 802.1Q **7-4**
Enabling NetWare Routing **7-4**
Defining the VLAN Encapsulation Format **7-4**
Configuring NetWare on the Subinterface **7-4**
IEEE 802.1Q Encapsulation Configuration Examples **7-5**
Configuring AppleTalk over IEEE 802.1Q Example **7-5**
Configuring IP Routing over IEEE 802.1Q Example **7-5**
Configuring IPX Routing over IEEE 802.1Q Example **7-5**
VLAN Commands **7-6**
clear vlan statistics **7-6**
debug vlan packet **7-6**
encapsulation dot1q **7-7**
show vlans **7-7**

GLOSSARY

INDEX

# About This Guide

This section discusses the objectives, audience, conventions, and organization of the *Cisco 1751 Router Software Configuration Guide* and provides general information about Cisco IOS software documentation.

Cisco documentation and additional literature are available in a CD-ROM package that ships with your product. The Documentation CD-ROM, a member of the Cisco Connection Family, is updated monthly. Therefore, it might be more up to date than printed documentation. To order additional copies of the Documentation CD-ROM, contact your local sales representative or call customer service. The CD-ROM package is available as a single package or as an annual subscription. You can also access Cisco documentation on the World Wide Web at http://www.cisco.com, http://www-china.cisco.com, or http://www-europe.cisco.com.

## Objectives

This guide describes the tasks and commands necessary to configure Voice-over-IP (VoIP) and virtual LANs (VLANs), and contains corresponding command-reference information for both topics.

## Audience

This publication is intended primarily for users who configure and maintain routers, but are not necessarily familiar with tasks, the relationship between tasks, or the commands necessary to perform particular tasks to configure VoIP. In addition, this publication is intended for users with some familiarity with IP and telephony networks.

## Cisco IOS Software Documentation

In addition to the information provided in this publication, you might need to refer to the Cisco IOS documentation set. The Cisco IOS software documentation is divided into nine modules and two master indexes. (See Figure 1.) Each module consists of two books: a configuration guide and a corresponding command reference. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference provide complete command syntax information. Each configuration guide can be used in conjunction with its corresponding command reference.

Two master indexes provide indexing information for the Cisco IOS software documentation set: an index for the configuration guides and an index for the command references. In addition, individual books contain a book-specific index.

***Figure 1 Cisco IOS Software Documentation Modules***

Module FC/FR:
**Configuration Fundamentals**
- Access Server and Router Product Overview
- Cisco IOS Software Configuration Basics
- Images and Configuration Files
- Interface Configuration
- System Management

Module P1C/P1R:
**Network Protocols, Part 1**
- IP Addressing
- IP Services
- IP Routing Protocols

Module P2C/P2R:
**Network Protocols, Part 2**
- AppleTalk
- Novell IPX

Module P3C/P3R:
**Network Protocols, Part 3**
- Apollo Domain
- Banyan VINES
- DECnet
- ISO CLNS
- XNS

Module WC/WR:
**Wide-Area Networking**
- ATM
- Frame Relay
- SMDS
- X.25 and LAPB

Module SC/SR:
**Security**
- Terminal Access Security
- Network Access Security
- Accounting and Billing
- Filtering Traffic
- Preventing Fraudulent Route Updates
- Network Data Encryption

Module DC/DR:
**Dial Solutions**
- Dial Business Solutions and Examples
- Dial-In Port Setup
- DDR and Dial Backup
- Remote Node and Terminal Service
- Cost-Control and Large-Scale Dial Solutions
- VPDN

Module XC/XR:
**Cisco IOS Switching Services**
- Switching Paths for IP Networks
  - Fast Switching
  - Autonomous Switching
  - NetFlow Switching
  - Optimum Switching
- Virtual LAN (VLAN) Switching and Routing
  - Inter-Switch Link Protocol Encapsulation
  - IEEE 802.10 Encapsulation
  - LAN Emulation

Module BC/BR:
**Bridging and IBM Networking**
- Transparent Bridging
- Source-Route Bridging
- Remote Source-Route Bridging
- DLSw+
- STUN and BSTUN
- LLC2 and SDLC
- IBM Network Media Translation
- DSPU and SNA Service Point
- SNA Frame Relay Access Support
- APPN
- NCIA Client/Server Topologies
- IBM Channel Attach

**Configuration Guide Master Index**

**Command Reference Master Index**

# Organization

Table 1 describes the contents of each chapter in this document.

***Table 1 Organization***

| Chapter | Title | Description |
|---|---|---|
| Chapter 1 | Voice over IP Overview | Overview of the VoIP software application and, for those unfamiliar with telephony, a brief Voice Primer. |
| Chapter 2 | VoIP Configuration | A general description of VoIP, necessary prerequisite tasks, configuration procedures for VoIP (including verification and troubleshooting tips), suggestions for optimizing dial peer and network interface configurations, and a discussion of how to configure Frame Relay and Microsoft NetMeeting to work with VoIP. |
| Chapter 3 | VoIP Configuration Examples | Four scenario-based VoIP configuration examples. |
| Chapter 4 | VoIP Commands | An alphabetical list of the Cisco IOS software commands used to configure VoIP. |
| Chapter 5 | VoIP Debug Commands | An alphabetical list of the Cisco IOS software debug commands used in conjunction with VoIP. |
| Chapter 6 | Routing Between Virtual LANs Overview | Overview of VLANs and routing between VLANs. |
| Chapter 7 | Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation | A general description of how to configure routing between VLANs using IEEE 802.1Q encapsulation and an alphabetical list of supported Cisco IOS software commands used to configure VLANs. |

# Command Syntax Conventions

Table 2 describes the syntax used with the commands in this document.

***Table 2 Command Syntax Guide***

| Convention | Description |
|---|---|
| **boldface** | Commands and keywords. |
| *italic* | Command input that is supplied by you. |
| [ ] | Keywords or arguments that appear within square brackets are optional. |
| { x \| x \| x } | A choice of keywords (represented by x) appears in braces separated by vertical bars. You must select one. |
| ^ or Ctrl | Represent the key labeled *Control*. For example, when you read ^*D* or *Ctrl-D*, you should hold down the Control key while you press the D key. |
| `screen font` | Examples of information displayed on the screen. |

*Table 2 Command Syntax Guide*

| Convention | Description |
|---|---|
| **boldface screen font** | Examples of information that you must enter. |
| < > | Nonprinting characters, such as passwords, appear in angled brackets. |
| [ ] | Default responses to system prompts appear in square brackets. |

# Cisco Connection Online

Cisco Connection Online (CCO) is Cisco Systems' primary, real-time support channel. Maintenance customers and partners can self-register on CCO to obtain additional information and services.

Available 24 hours a day, 7 days a week, CCO provides a wealth of standard and value-added services to Cisco's customers and business partners. CCO services include product information, product documentation, software updates, release notes, technical tips, the Bug Navigator, configuration notes, brochures, descriptions of service offerings, and download access to public and authorized files.

CCO serves a wide variety of users through two interfaces that are updated and enhanced simultaneously: a character-based version and a multimedia version that resides on the World Wide Web (WWW). The character-based CCO supports Zmodem, Kermit, Xmodem, FTP, and Internet e-mail, and it is excellent for quick access to information over lower bandwidths. The WWW version of CCO provides richly formatted documents with photographs, figures, graphics, and video, as well as hyperlinks to related information.

You can access CCO in the following ways:

- WWW: http://www.cisco.com
- WWW: http://www-europe.cisco.com
- WWW: http://www-china.cisco.com
- Telnet: cco.cisco.com
- Modem: From North America, 408 526-8070; from Europe, 33 1 64 46 40 82. Use the following terminal settings: VT100 emulation; databits: 8; parity: none; stop bits: 1; and connection rates up to 28.8 kbps.

For a copy of CCO's Frequently Asked Questions (FAQ), contact cco-help@cisco.com. For additional information, contact cco-team@cisco.com.

**Note** If you are a network administrator and need personal technical assistance with a Cisco product that is under warranty or covered by a maintenance contract, contact Cisco's Technical Assistance Center (TAC) at 800 553-2447, 408 526-7209, or tac@cisco.com. To obtain general information about Cisco Systems, Cisco products, or upgrades, contact 800 553-6387, 408 526-7208, or cs-rep@cisco.com.

# Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit comments electronically. Click **Feedback** on the toolbar, and then select **Documentation**. After you complete the form, click **Submit** to send it to Cisco. We appreciate your comments.

# 1

# Voice over IP Overview

Voice over IP (VoIP) enables a Cisco 1751 router (hereafter referred to as the router) to carry voice traffic (for example, telephone calls and faxes) over an IP network. Cisco's voice support is implemented using voice packet technology. In VoIP, the digital signal processor (DSP) segments the voice signal into frames and stores them in voice packets. These voice packets are transported using IP in compliance with the International Telecommunications Union-Telecommunications (ITU-T) specification H.323, the specification for transmitting multimedia (voice, video, and data) across a network. Because it is a delay-sensitive application, you need to have a well-engineered, end-to-end network to successfully use VoIP. Fine-tuning your network to adequately support VoIP involves a series of protocols and features to improve quality of service (QoS). Traffic shaping considerations must also be taken into account to ensure the reliability of the voice connection.

VoIP is primarily a software feature; however, you must install the voice interface cards (VICs) in the router. For more information about installing a VIC in the router, refer to the *Cisco WAN Interface Cards Hardware Installation Guide*.

## Voice Primer

The Voice Primer section provides supplementary information for those users unfamiliar with voice telephony. To understand Cisco's voice implementations, it helps to have some understanding of the analog and digital transmission and signaling. This section provides some very basic, abbreviated voice telephony information as background to help you configure VoIP, Voice over Frame Relay, Voice over ATM, and Voice over HDLC and contains the following topics:

- How VoIP Processes a Typical Telephone Call
- Numbering Scheme
- Analog Compared with Digital
- CODECs
- Delay
- Echo
- Signaling

## How VoIP Processes a Typical Telephone Call

Before configuring VoIP on your router, it helps to understand what happens at an application level when you place a call using VoIP. The general flow of a two-party voice call using VoIP is as follows:

1. The user picks up the handset; this signals an off-hook condition to the signaling application part of VoIP in the router.
2. The session application part of VoIP issues a dial tone and waits for the user to dial a telephone number.
3. The user dials the telephone number; those numbers are accumulated and stored by the session application.
4. After enough digits are accumulated to match a configured destination pattern, the telephone number is mapped to an IP host via the dial plan mapper. The IP host has a direct connection to either the destination telephone number or a PBX that is responsible for completing the call to the configured destination pattern.
5. The session application then runs the H.323 session protocol to establish a transmission and a reception channel for each direction over the IP network. If the call is being handled by a Private Branch Exchange (PBX), the PBX forwards the call to the destination telephone. If Resource Reservation Protocol (RSVP) has been configured, the RSVP reservations are put into effect to achieve the desired QoS over the IP network.
6. The coder-decoder compression schemes (CODECs) are enabled for both ends of the connection and the conversation proceeds using Real-Time Transport Protocol/User Datagram Protocol/Internet Protocol (RTP/UDP/IP) as the protocol stack.
7. Any call-progress indications (or other signals that can be carried inband) are cut through the voice path as soon as end-to-end audio channel is established. Signaling that can be detected by the voice ports (for example, inband dual-tone multifrequency (DTMF) digits after the call setup is complete) is also trapped by the session application at either end of the connection and carried over the IP network encapsulated in Real-Time Transport Control Protocol (RTCP) using the RTCP application-defined (APP) extension mechanism.
8. When either end of the call hangs up, the RSVP reservations are torn down (if RSVP is used) and the session ends. Each end becomes idle, waiting for the next off-hook condition to trigger another call setup.

## Numbering Scheme

The standard PSTN is a large, circuit-switched network. It uses a specific numbering scheme, which complies with the ITU-T **international public telecommunications numbering plan** (E.164) recommendations. For example, in North America, the North American Numbering Plan (NANP) is used, which consists of an area code, an office code, and a station code. Area codes are assigned geographically, office codes are assigned to specific switches, and station codes identify a specific port on that switch. The format in North America is 1Nxx-Nxx-xxxx, with N = digits 2 through 9 and x = digits 0 through 9. Internationally, each country is assigned a one- to three-digit country code; the country's dialing plan follows the country code. In Cisco's voice implementations, numbering schemes are configured using the **destination-pattern** command.

## Analog Compared with Digital

Analog transmission is not particularly robust or efficient at recovering from line noise. Because analog signals degrade over distance, they need to be periodically amplified; this amplification boosts both the voice signal and ambient line noise, resulting in degradation of the quality of the transmitted sound.

In response to the limitations of analog transmission, the telephony network migrated to digital transmission using pulse code modulation (PCM) or adaptive differential PCM (ADPCM). In both cases, analog sound is converted into digital form by sampling the analog sound 8000 times per second and converting each sample into a numeric code.

## CODECs

Pulse code modulation (PCM) and adaptive differential PCM (ADPCM) are examples of "waveform" CODEC techniques. Waveform CODECs are compression techniques that exploit the redundant characteristics of the waveform itself. In addition to waveform CODECs, there are source CODECs that compress speech by sending only simplified parametric information about voice transmission; these CODECs require less bandwidth. Source CODECs include linear predictive coding (LPC), code-excited linear prediction (CELP) and multipulse-multilevel quantization (MP-MLQ).

Coding techniques for telephony and voice packet are standardized by the ITU-T in its G-series recommendations. The Cisco 1751 router uses the following coding standards:

- G.711—Describes the 64-kbps PCM voice coding technique. In G.711, encoded voice is already in the correct format for digital voice delivery in the PSTN or through PBXs.
- G.729—Describes CELP compression where voice is coded into 8-kbps streams. There are two variations of this standard (G.729 and G.729 Annex A) that differ mainly in computational complexity; both provide speech quality similar to 32-kbps ADPCM.
- G.723—Describes a compression technique that can be used for compressing speech or audio signal components at very low bit rate as part of the H.324 family of standards. This CODEC has two bit rates associated with it: 5.3 kbps and 6.3 kbps. The higher bit rate is based on ML-MLQ technology and provides a somewhat higher quality of sound. The lower bit rate is based on CELP and provides system designers with additional flexibility.
- G.726—Describes ADPCM coding at 40, 32, 24, and 16 kbps. ADPCM-encoded voice can be interchanged between packet voice, PSTN, and PBX networks if the PBX networks are configured to support ADPCM.

In Cisco's voice implementations, compression schemes are configured using the **codec** command.

### Mean Opinion Score

Each CODEC provides a certain quality of speech. The quality of transmitted speech is a subjective response of the listener. A common benchmark used to determine the quality of sound produced by specific CODECs is the mean opinion score (MOS). With MOS, a wide range of listeners judge the quality of a voice sample (corresponding to a particular CODEC) on a scale of 1 (bad) to 5 (excellent). The scores are averaged to provide the MOS for that sample. Table 1-1 shows the relationship between CODECs and MOS scores.

*Table 1 Compression Methods and MOS Scores*

| Compression Method | Bit Rate (kbps) | MOS Score |
|---|---|---|
| G.711 PCM | 64 | 4.1 |
| G.723.1 MP-MLQ | 6.3 | 3.9 |
| G.723.1 ACELP | 5.3 | 3.65 |
| G.726 ADPCM | 32 | 3.85 |
| G.729 CS-ACELP[1] | 8 | 3.92 |
| G.729 x 2 Encodings | 8 | 3.27 |
| G.729 x 3 Encodings | 8 | 2.68 |
| G.729a CS-ACELP | 8 | 3.7 |

1. Conjugate structure-algebraic code-excited linear prediction

Although it might seem logical from a financial standpoint to convert all calls to low bit-rate CODECs to save on infrastructure costs, you should exercise additional care when designing voice networks with low bit-rate compression. There are drawbacks to compressing voice. One of the main drawbacks is signal distortion due to multiple encodings (called tandem encodings). For example, when a G.729 voice signal is tandem-encoded three times, the MOS score drops from 3.92 (very good) to 2.68 (unacceptable). Another drawback is CODEC-induced delay with low bit-rate CODECs.

# Delay

One of the most important design considerations in implementing voice is minimizing one-way, end-to-end delay. Voice traffic is real-time traffic; if there is too long a delay in voice packet delivery, speech will be unrecognizable. Delay is inherent in voice-networking and is caused by a number of different factors. An acceptable delay is less than 200 milliseconds.

There are basically two kinds of delay inherent in today's telephony networks: propagation delay and handling delay. Propagation delay is caused by the characteristics of the speed of light traveling via a fiber-optic-based or copper-based medium. Handling delay (sometimes called serialization delay) is caused by the devices that handle voice information. Handling delays have a significant impact on voice quality in a packet network.

CODEC-induced delays are considered a handling delay. Table 1-2 shows the delay introduced by different CODECs.

*Table 2 CODEC-Induced Delays*

| CODEC | Bit Rate (kbps) | Compression Delay (ms) |
|---|---|---|
| G.711 PCM | 64 | 5 |
| G.723.1 MP-MLQ | 6.3 | 30 |
| G.723.1 ACELP | 5.3 | 30 |
| G.726 ADPCM | 32 | 1 |

*Table 2 CODEC-Induced Delays*

| CODEC | Bit Rate (kbps) | Compression Delay (ms) |
|---|---|---|
| G.729 CS-ACELP | 8 | 15 |
| G.729a CS-ACELP | 8 | 15 |

Another handling delay is the time it takes to generate a voice packet. In VoIP, the DSP generates a frame every 10 milliseconds. Two of these frames are then placed within one voice packet; the packet delay is therefore 20 milliseconds.

Another source of handling delay is the time it takes to move the packet to the output queue. Cisco IOS software expedites the process of determining packet destination and getting the packet to the output queue. The actual delay at the output queue is another source of handling delay and should be kept under 10 milliseconds whenever possible by using whatever queuing methods are optimal for your network. Output queue delays are a QoS issue in VoIP and are discussed in the "Configure IP Networks for Real-Time Voice Traffic" section on page 2-2.

In Voice over Frame Relay, you need to make sure that voice traffic is not crowded out by data traffic. Strategies on how to manage Voice-over-Frame-Relay voice traffic are discussed in the "Configure Frame Relay for VoIP" section on page 2-24.

### Jitter

Jitter is another factor that affects delay. Jitter occurs when there is a variation between when a voice packet is expected to be received and when it actually is received, causing a discontinuity in the real-time voice stream. Voice devices such as the Cisco 3600 router, Cisco MC3810, and the Cisco 1751 router compensate for jitter by setting up a playout buffer to playback voice in a smooth fashion. Playout control is handled through RTP encapsulation, either by selecting adaptive or non-adaptive playout-delay mode. In either mode, the default value for nominal delay is sufficient.

### End-to-End Delay

Figuring out the end-to-end delay is not difficult if you know the end-to-end signal paths/data paths, the CODEC, and the payload size of the packets. Adding the delays from the end points to the CODECs at both ends, the encoder delay (which is 5 milliseconds for the G.711 and G.726 CODECs and 10 milliseconds for the G.729 CODEC), the packet delay, and the fixed portion of the network delay yields the end-to-end delay for the connection.

## Echo

Echo is hearing your own voice in the telephone receiver while you are talking. When timed properly, echo is reassuring to the speaker; if the echo exceeds approximately 25 milliseconds, it can be distracting and cause breaks in the conversation. In a traditional telephony network, echo is normally caused by a mismatch in impedance from the four-wire network switch conversion to the two-wire local loop and controlled by echo cancellers. In voice-packet based networks, echo cancellers are built into the low bit-rate CODECs and are operated on each DSP. Echo cancellers are limited by design by the total amount of time they will wait for the reflected speech to be received, which is known as an echo trail. The echo trail is normally 32 milliseconds.

In Cisco's voice implementations, echo cancellers are enabled using the **echo-cancel enable** command. The echo trails are configured using the **echo-cancel-coverage** command. VoIP has configurable echo trails of 8, 16, 24, and 32 milliseconds.

## Signaling

Although there are various types of signaling used in telecommunications today, this document describes only those with direct applicability to Cisco's voice implementations. The first one involves access signaling, which determines when a line has gone off-hook or on-hook (in other words, dial tone). FXS and FXO are types of access signaling. There are two common methods of providing this basic signal:

- Loop start is the most common technique for access signaling in a standard PSTN end-loop network. When a handset is picked-up (goes off-hook), this action closes the circuit that draws current from the telephone company's central office (CO), indicating a change in status. This change in status signals the CO to provide a dial tone. An incoming call is signalled from the CO to the handset by sending a signal in a standard on/off pattern, which causes the telephone to ring.
- Ground start is another access signaling method used to indicate on-hook/off-hook status to the CO, but this signaling method is primarily used on trunk lines or tie-lines between PBXs. Ground-start signaling works by using ground and current detectors. This allows the network to indicate off-hook or seizure of an incoming call independent of the ringing signal.

In Cisco's voice implementations, access signaling is configured using the **signal** command.

Another signaling technique used mainly between PBXs or other network-to-network telephony switches is known as E&M. There are five types of E&M signaling, as well as two different wiring methods. Cisco's voice implementation supports E&M types I, II, III, and V, using both two-wire and four-wire implementations. In Cisco's voice implementations, E&M signal types are configured using the **type** command.

# 2

# VoIP Configuration

This chapter explains how to configure VoIP on your router and contains the following sections:

- Prerequisite Tasks
- Configuration Tasks
- Configure IP Networks for Real-Time Voice Traffic
- Configure Number Expansion
- Configure Dial Peers
- Configure Voice Ports
- Additional VoIP Dial Peer Configurations
- Configure Frame Relay for VoIP
- Configure Microsoft NetMeeting for VoIP

## Prerequisite Tasks

Before you can configure your router to use VoIP, you need to perform the following tasks:

- Establish a working IP network. For more information about configuring IP, refer to the "IP Overview," "Configuring IP Addressing," and "Configuring IP Services" chapters in the Network Protocols Configuration Guide, Part 1 for Cisco IOS Release 12.1T.
- Install the voice interface cards (VICs) in your router. For more information about installing a VIC in your router, refer to the *Cisco WAN Interface Cards Hardware Installation Guide*.
- Complete your company's dial plan.
- Establish a working telephony network based on your company's dial plan.
- Integrate your dial plan and telephony network into your existing IP network topology. Merging your IP and telephony networks depends on your particular IP and telephony network topology. In general, we recommend the following:
  - Use canonical numbers wherever possible. Avoid situations where numbering systems are significantly different on different routers or access servers in your network.

- Make routing and dialing transparent to the user—for example, avoid secondary dial tones from secondary switches, where possible.
- Contact your PBX vendor for instructions about how to reconfigure the appropriate PBX interfaces.

After you have analyzed your dial plan and decided how to integrate it into your existing IP network, you are ready to configure your network devices to support VoIP.

# Configuration Tasks

To configure VoIP on your router, you need to perform the following steps:

**Step 1** Configure your IP network to support real-time voice traffic. Refer to the following section for information about selecting and configuring the appropriate QoS tool or tools to optimize voice traffic on your network.

**Step 2** (Optional) If you plan to run VoIP over Frame Relay, you need to consider certain factors so that VoIP runs smoothly. For example, a public Frame Relay cloud provides no guarantees for QoS. Refer to the "Configure Frame Relay for VoIP" section on page xxiv for information about deploying VoIP over Frame Relay.

**Step 3** Use the **num-exp** command to configure number expansion if your telephone network is configured so that you can reach a destination by dialing only a portion (an extension number) of the full E.164 telephone number. Refer to the "Configure Number Expansion" section on page viii for information about number expansion.

**Step 4** Use the **dial-peer voice** command to define dial peers and switch to the dial-peer configuration mode. Refer to the "Configure Dial Peers" section on page ix and the "Additional VoIP Dial Peer Configurations" section on page xxi for additional information about configuring dial peers and dial-peer characteristics.

**Step 5** Configure your router to support voice ports. Refer to the "Configure Voice Ports" section on page xiv for information about configuring voice ports.

# Configure IP Networks for Real-Time Voice Traffic

You need to have a well-engineered, end-to-end network when running delay-sensitive applications such as VoIP. Fine-tuning your network to adequately support VoIP involves a series of protocols and features to improve QoS. It is beyond the scope of this document to explain the specific details relating to wide-scale QoS deployment. Cisco IOS software provides many tools for enabling QoS on your backbone, such as Random Early Detection (RED), Weighted Random Early Detection (WRED), Fancy Queuing (meaning custom, priority, or weighted fair queuing), and IP precedence. To configure your IP network for real-time voice traffic, you need to take into consideration the entire scope of your network and then select the appropriate QoS tool or tools.

The important thing to remember is that QoS must be configured throughout your network—not just on your router running VoIP—to improve voice network performance. Not all QoS techniques are appropriate for all network routers. Edge routers and backbone routers in your network do not necessarily perform the same operations; the QoS tasks they perform might differ as well. To configure your IP network for real-time voice traffic, you need to consider the functions of both edge and backbone routers in your network and then select the appropriate QoS tool or tools.

In general, edge routers perform the following QoS functions:

- Packet classification
- Admission control
- Bandwidth management
- Queuing

In general, backbone routers perform the following QoS functions:

- High-speed switching and transport
- Congestion management
- Queue management

Scalable QoS solutions require cooperative edge and backbone functions.

Although not mandatory, some QoS tools can be valuable in fine-tuning your network to support real-time voice traffic. To configure your IP network for QoS, perform one or more of the following tasks:

- Configure RSVP for Voice
- Configure Multilink PPP with Interleaving
- Configure RTP Header Compression
- Configure Custom Queuing
- Configure Weighted Fair Queuing

Each of these tasks is discussed in the following sections.

# Configure RSVP for Voice

Resource Reservation Protocol (RSVP) enables routers to reserve enough bandwidth on an interface for reliability and quality performance. RSVP allows end systems to request a particular QoS from the network. Real-time voice traffic requires network consistency. Without consistent QoS, real-time traffic can experience jitter, insufficient bandwidth, delay variations, or information loss. RSVP works in conjunction with current queuing mechanisms. It is up to the interface queuing mechanism (such as weighted fair queuing or WRED) to implement the reservation.

RSVP works well on PPP, HDLC, and similar serial line interfaces. It does not work well on multi-access LANs. RSVP can be equated to a dynamic access list for packet flows.

You should configure RSVP to ensure QoS if the following conditions describe your network:

- Small scale voice network implementation
- Links slower than 2 Mbps
- Links with high utilization
- Need for the best possible voice quality

## Enable RSVP

To minimally configure RSVP for voice traffic, you must enable RSVP on each interface where priority needs to be set.

By default, RSVP is disabled so that it is backwards compatible with systems that do not implement RSVP. To enable RSVP for IP on an interface, use the following interface configuration command:

```
Router(config-if)# ip rsvp bandwidth [interface-kbps] [single-flow-kbps]
```

This command starts RSVP and sets the bandwidth and single-flow limits. The default maximum bandwidth is up to 75 percent of the bandwidth available on the interface. By default, the amount reservable by a flow can be up to the entire reservable bandwidth.

On subinterfaces, RSVP applies to the more restrictive of the available bandwidths of the physical interface and the subinterface.

Reservations on individual circuits that do not exceed the single flow limit normally succeed. However, if reservations have been made on other circuits adding up to the line speed, and a reservation is made on a subinterface that itself has enough remaining bandwidth, it will still be refused because the physical interface lacks supporting bandwidth.

A Cisco 1751 router running VoIP and configured for RSVP requests allocations using the following formula:

```
bps=packet_size+ip/udp/rtp header size * 50 per second
```

For G.729, the allocation works out to be 24,000 bps. For G.711, the allocation is 80,000 bps.

For more information about configuring RSVP, refer to the "Configuring RSVP" chapter of the *Network Protocols Configuration Guide, Part 1* for Cisco IOS Release 12.1T.

### RSVP Configuration Example

The following example enables RSVP and sets the maximum bandwidth to 100 kbps and the maximum bandwidth per single request to 32 kbps (the example presumes that both VoIP dial peers have been configured):

```
Router(config)# interface serial 0/0
Router(config-if)# ip rsvp bandwidth 100 32
Router(config-if)# fair-queue
Router(config-if)# end
```

After enabling RSVP, you must also use the **req-qos** dial-peer configuration command to request an RSVP session on each VoIP dial peer. Otherwise, no bandwidth is reserved for voice traffic.

```
Router(config)# dial-peer voice 211 voip
Router(config-dial-peer)# req-qos controlled-load

Router(config)# dial-peer voice 212 voip
Router(config-dial-peer)# req-qos controlled-load
```

## Configure Multilink PPP with Interleaving

Multiclass multilink PPP interleaving allows large packets to be multilink-encapsulated and fragmented into smaller packets to satisfy the delay requirements of real-time voice traffic; small real-time packets, which are not multilink-encapsulated, are transmitted between fragments of the large packets. The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be transmitted earlier than other flows. Interleaving provides the delay bounds for delay-sensitive voice packets on a slow link that is used for other best-effort traffic.

In general, multilink PPP with interleaving is used in conjunction with weighted fair queuing and RSVP or IP precedence to ensure voice packet delivery. Use multilink PPP with interleaving and weighted fair queuing to define how data is managed; use RSVP or IP precedence to give priority to voice packets.

You should configure multilink PPP if the following conditions describe your network:

- Point-to-point connection using PPP encapsulation

- Links slower than 2 Mbps

**Note** Do not use multilink PPP on links greater than 2 Mbps.

Multilink PPP support for interleaving can be configured on virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces. To configure interleaving, you need to complete the following tasks:

- Configure the dialer interface or virtual template, as defined in the relevant chapters of the *Dial Solutions Configuration Guide* for Cisco IOS Release 12.1T.
- Configure multilink PPP and interleaving on the interface or template.

To configure multilink PPP and interleaving on a configured and operational interface or virtual interface template, use the following interface configuration commands:

| Step | Command | Task |
|---|---|---|
| 1. | **ppp multilink** | Enable Multilink PPP. |
| 2. | **ppp multilink interleave** | Enable real-time packet interleaving. |
| 3. | **ppp multilink fragment-delay** *milliseconds* | Optionally, configure a maximum fragment delay of 20 milliseconds. |
| 4. | **ip rtp reserve** *lowest-UDP-port range-of-ports* [*maximum-bandwidth*] | Reserve a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows. This only applies if you have not configured RSVP. |

**Note** You can use the **ip rtp reserve** command instead of configuring RSVP. If you configure RSVP, this command is not required.

For more information about multilink PPP, refer to the "Configuring Media-Independent PPP and Multilink PPP" chapter in the *Dial Solutions Configuration Guide* for Cisco IOS Release 12.1T.

## Multilink PPP Configuration Example

The following example defines a virtual interface template that enables multilink PPP with interleaving and a maximum real-time traffic delay of 20 milliseconds and then applies that virtual template to the multilink PPP bundle:

```
Router(config)# interface virtual-template 1
Router(config-if)# ppp multilink
Router(config-if)# encapsulated ppp
Router(config-if)# ppp multilink interleave
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ip rtp reserve 16384 100 64

Router(config)# multilink virtual-template 1
```

# Configure RTP Header Compression

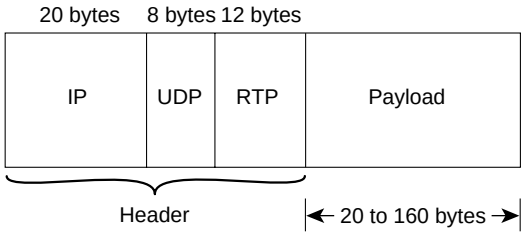
Real-Time Transport Protocol (RTP) is used for carrying audio traffic in packets over an IP network. RTP header compression compresses the IP/UDP/RTP header in an RTP data packet from 40 bytes to approximately 2 to 4 bytes (most of the time), as shown in Figure 1.

This compression feature is beneficial if you are running VoIP over slow links. Enabling compression on both ends of a low-bandwidth serial link can greatly reduce the network overhead if there is a lot of RTP traffic on that slow link.

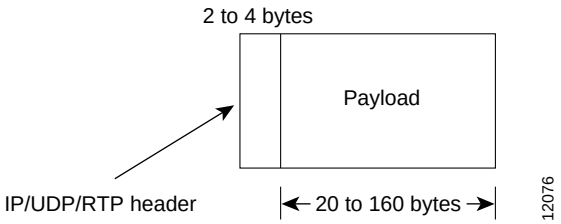
Typically, an RTP packet has a payload of approximately 20 to 160 bytes for audio applications that use compressed payloads. RTP header compression is especially beneficial when the RTP payload size is small (for example, compressed audio payloads between 20 and 50 bytes).

***Figure 1 RTP Header Compression***

**Before RTP header compression:**




**After RTP header compression:**




You should configure RTP header compression if the following conditions describe your network:

- Links slower than 2 Mbps
- Need to save bandwidth

**Note** Do not use RTP header compression on links greater than 2 Mbps.

Perform the following tasks to configure RTP header compression for VoIP. The first task is required; the second task is optional.

- Enable RTP Header Compression on a Serial Interface
- Change the Number of Header Compression Connections

### Enable RTP Header Compression on a Serial Interface

You need to enable compression on both ends of a serial connection. To enable RTP header compression, use the following interface configuration command:

```
Router(config-if)# ip rtp header-compression [passive]
```

If you include the **passive** keyword, the software compresses outgoing RTP packets only if incoming RTP packets on the same interface are compressed. If you use the command without the **passive** keyword, the software compresses all RTP traffic.

### Change the Number of Header Compression Connections

By default, the software supports a total of 16 RTP header compression connections on an interface. To specify a different number of RTP header compression connections, use the following interface configuration command:

```
Router(config-if)# ip rtp compression connections number
```

### RTP Header Compression Configuration Example

The following example enables RTP header compression for a serial interface:

```
Router(config)# interface serial0
Router(config-if)# ip rtp header-compression
Router(config-if)# encapsulation ppp
Router(config-if)# ip rtp compression-connections 25
```

For more information about RTP header compression, see the “Configuring IP Multicast Routing” chapter of the Network Protocols Configuration Guide, Part 1 for Cisco IOS Release 12.1T.

## Configure Custom Queuing

Some QoS features, such as IP RTP reserve and custom queuing, are based on the transport protocol and the associated port number. Real-time voice traffic is carried on UDP ports ranging from 16384 to 16624. This number is derived from the following formula:

```
16384 + (4 x number of voice ports in the router)
```

Custom Queuing and other methods for identifying high priority streams should be configured for these port ranges. For more information about custom queuing, refer to the “Managing System Performance” chapter in the *Configuration Fundamentals Configuration Guide* for Cisco IOS Release 12.1T.

## Configure Weighted Fair Queuing

Weighted fair queuing ensures that queues do not starve for bandwidth and that traffic gets predictable service. Low-volume traffic streams receive preferential service; high-volume traffic streams share the remaining capacity, obtaining equal or proportional bandwidth.

In general, weighted fair queuing is used in conjunction with multilink PPP with interleaving and RSVP or IP precedence to ensure voice packet delivery. Use weighted fair queuing with multilink PPP to define how data is managed; use RSVP or IP precedence to give priority to voice packets. For more information about weighted fair queuing, refer to the “Managing System Performance” chapter in the *Configuration Fundamentals Configuration Guide* for Cisco IOS Release 12.1T.

# Configure Number Expansion

In most corporate environments, the telephone network is configured so that you can reach a destination by dialing only a portion (an extension number) of the full E.164 telephone number. VoIP can be configured to recognize extension numbers and expand them into their full E.164 dialed number by using two commands in tandem: **destination-pattern** and **num-exp**. Before you configure these two commands, it helps to map individual telephone extensions with their full E.164 dialed numbers. This can be done easily by creating a number expansion table.

## Create a Number Expansion Table

In Figure 2, a small company decides to use VoIP to integrate its telephony network with its existing IP network. The destination pattern (or expanded telephone number) associated with Cisco 1751 Router 1 (left of the IP cloud) is (408) 555-xxxx, where xxxx identifies the individual dial peers by extension. The destination pattern (or expanded telephone number) associated with Cisco 1751 Router 2 (right of the IP cloud) is (729) 555-xxxx.

***Figure 2 Sample VoIP Network***

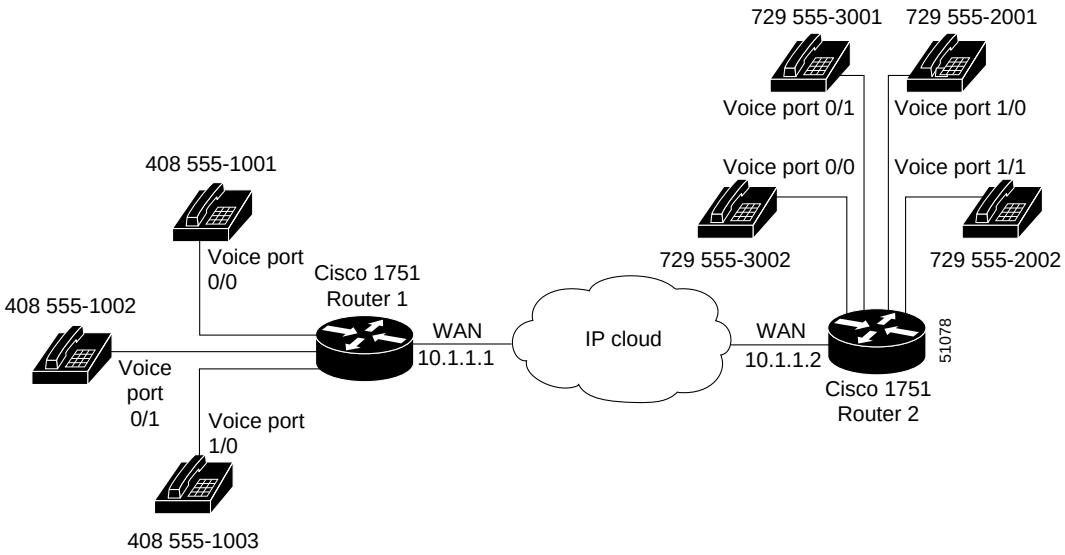



Table 1 shows the number expansion table for this scenario.

*Table 1 Sample Number Expansion Table*

| Extension | Destination Pattern | Num-Exp Command Entry | Description |
|---|---|---|---|
| 1... | 14085551... | num-exp 1... 14085551... | To expand a four-digit extension beginning with the numeral 1 by prefixing 1408555 to it |
| 2... | 17295552... | num-exp 2... 17295552... | To expand a four-digit extension beginning with the numeral 2 by prefixing 1408555 to it |
| 3... | 17295553... | num-exp 3... 17295553... | To expand a four-digit extension beginning with the numeral 3 by prefixing 1408555 to it |

**Note** You can use a period (.) to represent variables (such as extension numbers) in a telephone number. A period is similar to a wildcard, which matches any entered digit.

The information included in this example needs to be configured on both Cisco 1751 Router 1 and Cisco 1751 Router 2. In this configuration, Cisco 1751 Router 1 can call any number string that begins with the digits *17295552* or *17295553* to connect to Cisco 1751 Router 2. Similarly, Cisco 1751 Router 2 can call any number string that begins with the digits *14085551* to connect to Cisco 1751 Router 1.

## Configure Number Expansion

To define how to expand an extension number into a particular destination pattern, use the following global configuration command:

```
Router(config)# num-exp extension-number extension-string
```

Use the **show num-exp** command to verify that you have mapped the telephone numbers correctly.

After you have configured dial peers and assigned destination patterns to them, use the **show dialplan number** command to see how a telephone number maps to a dial peer.

# Configure Dial Peers

The key to understanding how VoIP functions is to understand dial peers. All of the voice technologies use dial peers to define the characteristics associated with a call leg. A call leg is a discrete segment of a call connection that lies between two points in the connection, as shown in Figure 3 and Figure 4. For instance, between a telephone and a router, a router and a network, a router and a PBX, or a router and the PSTN. Each call leg corresponds to a dial peer. An end-to-end call is comprised of four call legs, two from the perspective of the source router as shown in Figure 3, and two from the perspective of the destination router as shown in Figure 4. Dial peers are used to apply specific attributes to call legs and to identify call origin and destination. Attributes applied to a call leg include QoS, CODEC, voice activity detection (VAD), and fax rate.

*Figure 3 Dial Peer Call Legs from the Perspective of the Source Router*

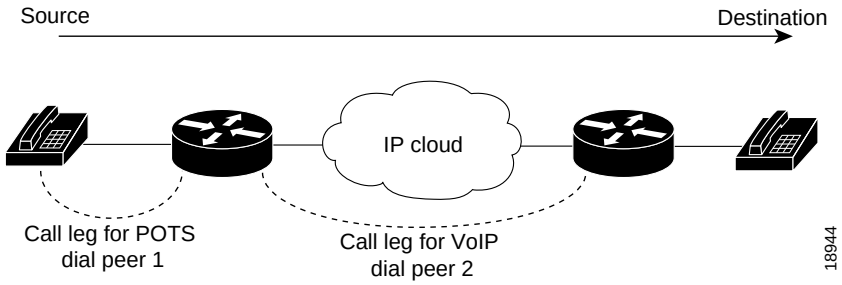
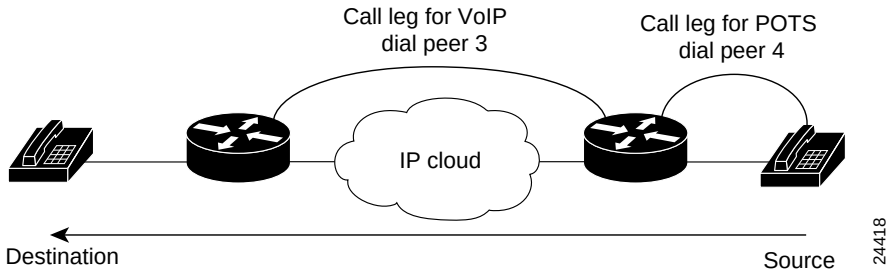



*Figure 4 Dial Peer Call Legs from the Perspective of the Destination Router*




There are basically two different kinds of dial peers with each voice implementation:

- POTS—(also known as “plain old telephone service” or “basic telephone service”) dial peer associates a physical voice port with a local telephone device, and the key commands you need to configure are the **port** and **destination-pattern** commands. The **destination-pattern** command defines the telephone number associated with the POTS dial peer. The **port** command associates the POTS dial peer with a specific logical dial interface, normally the voice port connecting your router to the local POTS network.
- VoIP—dial peer associates a telephone number with an IP address, and the key commands you need to configure are the **destination-pattern** and **session target** commands. The **destination-pattern** command defines the telephone number associated with the VoIP dial peer. The **session target** command specifies a destination IP address for the VoIP dial peer. In addition, you can use VoIP dial peers to define characteristics such as IP precedence, additional QoS parameters (when RSVP is configured), CODEC, and VAD.

# Inbound versus Outbound Dial Peers

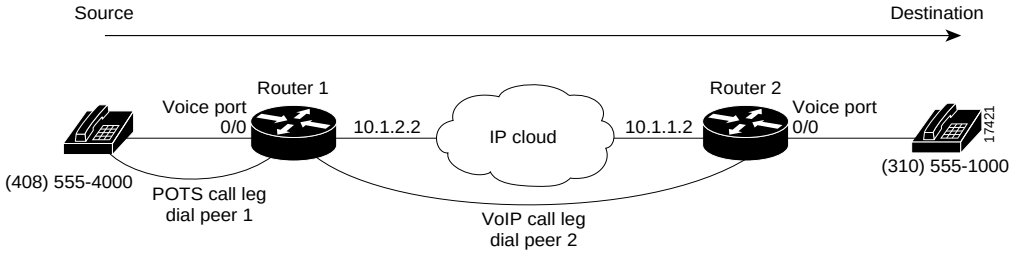
Dial peers are used for both inbound and outbound call legs. It is important to remember that these terms are defined from the *router* perspective. An inbound call leg means that an incoming call comes *to* the router. An outbound call leg means that an outgoing call is placed *from* the router.

For inbound call legs, a dial peer might be associated with the calling number or the voice-port number. Outbound call legs always have a dial peer associated with them. The destination pattern is used to identify the outbound dial peer. The call is associated with the outbound dial peer at setup time.

POTS dial peer associate a telephone number with a particular voice port so that incoming calls for that telephone number can be received and outgoing calls can be placed. VoIP dial peers point to specific devices (by associating destination telephone numbers with a specific IP address) so that incoming calls can be received and outgoing calls can be placed. Both POTS and VoIP dial peers are needed to establish VoIP connections.

Establishing communication using VoIP is similar to configuring an IP static route; you are establishing a specific voice connection between two defined endpoints. As shown in Figure 5, for outgoing calls (from the perspective of the POTS dial peer 1), the POTS dial peer establishes the source (via the originating telephone number or voice port) of the call. The VoIP dial peer establishes the destination by associating the destination telephone number with a specific IP address.

***Figure 5 Outgoing Calls from the Perspective of POTS Dial Peer 1***



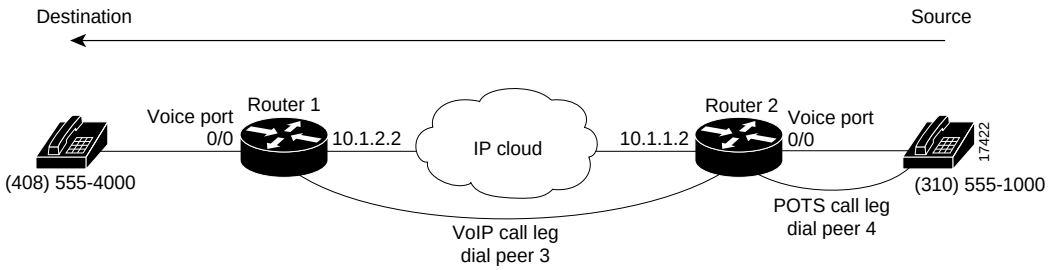

To configure call connectivity between the source and the destination as illustrated in Figure 5, enter the following commands on router 10.1.2.2:

```
Router(config)# dial-peer voice 1 pots
Router(config-dial-peer)# destination-pattern 14085554000
Router(config-dial-peer)# port 0/0

Router(config)# dial-peer voice 2 voip
Router(config-dial-peer)# destination-pattern 13105551000
Router(config-dial-peer)# session target ipv4:10.1.1.2
```

Figure 6 shows how to complete the end-to-end call between dial peer 1 and dial peer 4.

***Figure 6 Outgoing Calls from the Perspective of POTS Dial Peer 2***



To complete the end-to-end call between dial peer 1 and dial peer 4 as illustrated in Figure 6, enter the following commands on router 10.1.1.2:

```
Router(config)# dial-peer voice 4 pots
Router(config-dial-peer)# destination-pattern 13105551000
Router(config-dial-peer)# port 0/0

Router(config)# dial-peer voice 3 voip
Router(config-dial-peer)# destination-pattern 14085554000
Router(config-dial-peer)# session target ipv4:10.1.2.2
```

# Create a Dial-Peer Configuration Table

There is specific data relative to each dial peer that needs to be identified before you can configure dial peers in VoIP. One way to do this is to create a dial peer configuration table.

Using the example in Figure 2, Router 1, with an IP address of 10.1.1.1, connects a small sales branch office to the main office through Router 2. There are three telephones in the sales branch office that need to be established as dial peers. Router 2, with an IP address of 10.1.1.2, is the primary gateway to the main office. There are four devices that need to be established as dial peers in the main office, all of which are basic telephones connected to the PBX. Figure 2 on page 2-8 shows a diagram of this small voice network, and Table 1 shows the dial peer configuration table for the example in the figure.

***Table 2 Dial-Peer Configuration Table for Sample VoIP Network***

| | | Commands | | | | |
|---|---|---|---|---|---|---|
| **Router** | **Dial Peer Tag** | **Destination-Pattern** | **Type** | **Session Target** | **CODEC** | **QoS** |
| *Cisco 1751 Router 1* | 10 | 1729555.... | VoIP | IPV4 10.1.1.2 | G.729 | Best effort |
| *Cisco 1751 Router 2* | 11 | 1408555.... | VoIP | IPV4 10.1.1.1 | G.729 | Best effort |

# Configure POTS Dial Peers

POTS dial peers enable incoming calls to be received by a particular telephony device. To configure a POTS dial peer, you need to uniquely identify the dial peer (by assigning it a unique tag number), define its telephone numbers, and associate it with a voice port through which calls are established. Under most circumstances, the default values for the remaining dial peer configuration commands are sufficient to establish connections.

To enter the dial peer configuration mode (and select POTS as the method of voice-related encapsulation), use the following global configuration command:

```
Router(config)# dial-peer voice number pots
```

The *number* value of the **dial-peer voice pots** command is a tag that uniquely identifies the dial peer. (This number has local significance only.)

To configure the identified POTS dial peer, use the following dial peer configuration command:

```
Router(config-dial-peer)# destination-pattern string
```

The *string* value of the **destination-pattern command** is the destination telephone number associated with this POTS dial peer.

## Outbound Dialing on POTS Dial Peers

When a router receives a voice call, it selects an outbound dial peer by comparing the called number (the full E.164 telephone number) in the call information with the number configured as the destination pattern for the POTS dial peer. The router then removes the left-justified numbers corresponding to the destination pattern that matches the called number. If you have configured a prefix, the prefix is put in front of the remaining numbers, creating a dial string, which the router then dials. If all numbers in the destination pattern are removed, the user receives (depending on the attached equipment) a dial tone.

For example, suppose there is a voice call with the E.164 called number of *1(310) 767-2222*. If you configure a destination-pattern of *1310767* and a prefix of *9*, the router removes *1310767* from the E.164 telephone number, leaving the extension number of *2222*. It will then prefix *9*, to the front of the remaining numbers, so that the actual numbers dialed are *9, 2222*. The comma in this example means that the router will pause for one second between dialing the *9* and the *2* to allow for a secondary dial tone.

For additional POTS dial-peer configuration options, refer to the "VoIP Commands" chapter.

# Configure VoIP Dial Peers

VoIP dial peers enable outgoing calls to be made from a particular telephony device. To configure a VoIP dial peer, you need to identify the dial peer (by assigning it a unique tag number), define its destination telephone number, and define its destination IP address. As with POTS dial peers, under most circumstances the default values for the remaining dial peer configuration commands are adequate to establish connections.

To enter the dial peer configuration mode (and select VoIP as the method of voice-related encapsulation), use the following global configuration command:

```
Router(config)# dial-peer voice number voip
```

The *number* value of the **dial-peer voice voip** command is a tag that uniquely identifies the dial peer.

To configure the identified VoIP dial peer, use the following dial peer configuration commands

| | Command | Task |
|---|---|---|
| Step 1 | **destination-pattern** *string* | Define the destination telephone number associated with this VoIP dial peer. |
| Step 2 | **session target** {**ipv4:***destination-address* \| **dns:***host-name*} | Specify a destination IP address for this dial peer. |

For additional VoIP dial peer configuration options, refer to the "VoIP Commands" chapter. For examples of how to configure dial peers, refer to the "VoIP Configuration Examples" chapter.

## Verifying Your Configuration

You can check the validity of your dial peer configuration by performing the following tasks:

- If you have relatively few dial peers configured, you can use the **show dial-peer voice** command to verify that the data configured is correct. Use this command to display a specific dial peer or to display all configured dial peers.
- Use the **show dialplan number** command to show which dial peer is reached when a particular number is dialed.

## Troubleshooting Tips

If you are having trouble connecting a call and you suspect the problem is associated with the dial-peer configuration, you can try to resolve the problem by performing the following tasks:

- Ping the associated IP address to confirm connectivity. If you cannot successfully ping your destination, refer to the "Configuring IP" chapter in the Network Protocols Configuration Guide, Part 1 for Cisco IOS Release 12.1T.
- Use the **show dial-peer voice** command to verify that the operational status of the dial peer is up.
- Use the **show dialplan number** command on the local and remote routers to verify that the data is configured correctly on both.
- If you have configured number expansion, use the **show num-exp** command to check that the partial number on the local router maps to the correct full E.164 telephone number on the remote router.
- If you have configured a CODEC value, there can be a problem if the VoIP dial peers on either side of the connection have incompatible CODEC values. Make sure that both VoIP peers have been configured with the same CODEC value.

**Caution** If you are not familiar with Cisco IOS debug commands, you should read the "Using Debug Commands" section in the "VoIP Debug Commands" chapter before attempting any debugging.

- Use the **debug vpm spi** command to verify the output string the router dials is correct.
- Use the **debug cch323 rtp** command to check RTP packet transport.
- Use the **debug cch323 h225** command to check the call setup.

# Configure Voice Ports

Your router provides only analog voice ports for its implementation of VoIP. The type of signaling associated with these analog voice ports depends on the voice interface card (VIC) installed in the device.

Each VIC is specific to a particular signaling type; therefore, VICs determine the type of signaling for the voice ports. Voice-port commands define the characteristics associated with a particular voice-port signaling type.

The voice ports support three basic voice signaling types:

- FXS—The foreign exchange station interface uses a standard RJ-11 modular telephone cable to connect directly to a standard telephone, fax machine, PBXs, or similar device, and supplies ring, voltage, and dial tone to the station.
- FXO—The foreign exchange office interface uses a RJ-11 modular telephone cable to connect local calls to a PSTN central office or to PBX that does not support E&M signaling. This interface is used for off-premise extension applications.
- E&M—The E&M interface uses a RJ-45 telephone cable to connect remote calls from an IP network to PBX trunk lines (tie lines) for local distribution. It is a signaling technique for two-wire and four-wire telephone and trunk interfaces.

## Configure FXS or FXO Voice Ports

Under most circumstances, the default voice-port values are adequate to configure FXS and FXO ports to transport voice data over your existing IP network. However, if you need to change the default configuration for these voice ports, use the following commands beginning in privileged EXEC mode:

| | Command | Required or Optional | Task |
|---|---|---|---|
| **Step 1** | **configure terminal** | Required | Enter the global configuration mode. |
| **Step 2** | **voice-port** *slot-number/port* | Required | Identify the voice port you want to configure and enter the voice port configuration mode. |
| **Step 3** | **dial-type** {**dtmf** \| **pulse**} | Required | (For FXO ports only) Select the appropriate dial type for out-dialing. |
| **Step 4** | **signal** {**loop-start** \| **ground-start**} | Required | Select the appropriate signal type for this interface. |
| **Step 5** | **cptone** *country* | Required | Select the appropriate voice call progress tone for this interface.<br>The default for this command is **us**. For a list of supported countries, refer to Chapter 4, “VoIP Commands.” |
| **Step 6** | **ring frequency** {**25** \| **50**} | Required | (For FXS ports only) Select the ring frequency (in Hz) specific to the equipment attached to this voice port and appropriate to the country you are in. |
| **Step 7** | **ring number** *number* | Required | (For FXO ports only) Specify the maximum number of rings before answering a call. |
| **Step 8** | **connection plar** *string* | Optional | Specify the private line auto ringdown (PLAR) connection if this voice port is used for a PLAR connection. The *string* value specifies the destination telephone number. |

| | Command | Required or Optional | Task |
|---|---|---|---|
| Step 9 | **music-threshold** *number* | Optional | Specify the threshold (in dB) for on-hold music. Valid entries are from –70 to –30 decibels (dB). |
| Step 10 | **description** *string* | Optional | Attach descriptive text about this voice-port connection. |
| Step 11 | **comfort-noise** | Optional | If voice activity detection (VAD) is activated, specify that background noise is generated. |

### Verifying Your Configuration

You can check the validity of your voice-port configuration by performing the following tasks:

- Pick up the handset of an attached telephony device and listen for a dial tone.
- Check for DTMF detection if you have a dial tone. If the dial tone stops when you dial a digit, the voice port is configured properly.
- Use the **show voice port** command to verify that the data configured is correct.

### Troubleshooting Tips

If you are having trouble connecting a call and you suspect the problem is associated with the voice-port configuration, you can try to resolve the problem by performing the following tasks:

- Ping the associated IP address to confirm connectivity. If you cannot ping your destination, refer to the *Network Protocols Configuration Guide, Part 1* for Cisco IOS Release 12.1T.
- Use the **show voice port** command to make sure that the port is enabled. If the port is offline, use the **no shutdown** command.
- Make sure the VICs are correctly installed. For more information about installing a VIC in your router, refer to the *Cisco WAN Interface Cards Hardware Installation Guide*.

## Fine-Tune FXS and FXO Voice Ports

In most cases, the default values for voice-port tuning commands are sufficient. Depending on the specifics of your particular network, you might need to adjust voice parameters involving timing, input gain, and output attenuation for FXS or FXO voice ports. Collectively, these commands are referred to as voice-port tuning commands.

If you need to change the default tuning configuration for FXS and FXO voice ports, use the following commands beginning in privileged EXEC mode:

| | Command | Task | Valid Entries | Default Values |
|---|---|---|---|---|
| Step 1 | **configure terminal** | Enter the global configuration mode. | | |
| Step 2 | **voice-port** *slot-number/port* | Identify the voice port you want to configure, and enter the voice port configuration mode. | | |
| Step 3 | **input gain** *value* | Specify (in dB) the amount of gain to be inserted at the receiver side of the interface. | –6 to 14 dB | 0 dB |
| Step 4 | **output attenuation** *value* | Specify (in dB) the amount of attenuation at the transmit side of the interface. | 0 to 14 dB | 0 dB |
| Step 5 | **echo-cancel enable** | Enable echo-cancellation of voice that is sent out of the interface and received back on the same interface. | | |
| Step 6 | **echo-cancel coverage** *value* | Adjust the size (in milliseconds) of the echo-cancel. | 8, 16, 24, and 32 ms | 16 ms |
| Step 7 | **non-linear** | Enable nonlinear processing, which shuts off any signal if no near-end speech is detected. (Nonlinear processing is used with echo-cancellation.) | | |
| Step 8 | **timeouts initial** *seconds* | Specify the number of seconds the system will wait for the caller to input the first digit of the dialed digits. | 0 to 120 sec | 10 sec |
| Step 9 | **timeouts interdigit** *seconds* | Specify the number of seconds the system will wait (after the caller has input the initial digit) for the caller to input a subsequent digit. | 0 to 120 sec | 10 sec |
| Step 10 | **timing digit** *milliseconds* | If the voice-port dial type is DTMF, configure the DTMF digit signal duration. | 50 to 100 ms | 100 ms |
| Step 11 | **timing inter-digit** *milliseconds* | If the voice-port dial type is DTMF, configure the DTMF inter-digit signal duration. | 50 to 500 ms | 100 ms |

| | Command | Task | Valid Entries | Default Values |
|---|---|---|---|---|
| Step 12 | **timing pulse-digit** *milliseconds* | (FXO ports only) If the voice-port dial type is pulse, configure the pulse digit signal duration. | 10 to 20 ms | 20 ms |
| Step 13 | **timing pulse-inter-digit** *milliseconds* | (FXO ports only) If the voice-port dial type is pulse, configure the pulse inter-digit signal duration. | 100 to 1000 ms | 500 ms |

**Note** After you change any voice-port command, we recommend that you cycle the port by using the **shutdown** and **no shutdown** commands.

# Configure E&M Voice Ports

Unlike FXS and FXO voice ports, the default E&M voice-port parameters are not sufficient to enable voice and data transmission over your IP network. Because of the inherent complexities of PBX networks, E&M voice-port values must match those specified by the particular PBX device to which it is connected.

To configure E&M voice ports, use the following commands beginning in privileged EXEC mode:

| | Command | Required / Optional | Task |
|---|---|---|---|
| Step 1 | **configure terminal** | Required | Enter the global configuration mode. |
| Step 2 | **voice-port** *slot-number/port* | Required | Identify the voice port you want to configure, and enter the voice port configuration mode. |
| Step 3 | **dial-type** {**dtmf** \| **pulse**} | Required | Select the appropriate dial type for out-dialing. |
| Step 4 | **signal** {**wink-start** \| **immediate** \| **delay-dial**} | Required | Select the appropriate signal type for this interface. |
| Step 5 | **cptone** {**australia** \| **brazil** \| **china** \| **finland** \| **france** \| **germany** \| **japan** \| **northamerica** \| **unitedkingdom**} | Required | Select the appropriate voice call progress tone for this interface. |
| Step 6 | **operation** {**2-wire** \| **4-wire**} | Required | Select the appropriate cabling scheme for this voice port. |

| | Command | Required / Optional | Task |
|---|---|---|---|
| Step 7 | **type** {**1** \| **2** \| **3** \| **5**} | Required | Select the appropriate E&M interface type.<br>Type 1 is for the following lead configuration:<br>E—output, relay to ground<br>M—input, referenced to ground<br>Type 2 is for the following lead configuration:<br>E—output, relay to SG<br>M—input, referenced to ground<br>SB—feed for M, connected to –48V<br>SG—return for E, galvanically isolated from ground<br>Type 3 is for the following lead configuration:<br>E—output, relay to ground<br>M—input, referenced to ground<br>SB—connected to –48V<br>SG—connected to ground<br>Type 5 is for the following lead configuration:<br>E—output, relay to ground<br>M—input, referenced to –48V. |
| Step 8 | **impedance** {**600c** \| **600r** \| **900c** \| **complex1** \| **complex2**} | Required | Specify a terminating impedance for an E&M voice port. The impedance value selected must match the specifications from the telephony system to which this voice port is connected. |
| Step 9 | **connection plar** *string* | Optional | Specify the private line auto ringdown (PLAR) connection if this voice port is used for a PLAR connection. The *string* value specifies the destination telephone number. |
| Step 10 | **music-threshold** *number* | Optional | Specify the threshold (in dB) for on-hold music. Valid entries are from –70 to –30 dB. The default is –38 dB. |
| Step 11 | **description** *string* | Optional | Attach descriptive text about this voice-port connection. |
| Step 12 | **comfort-noise** | Optional | Specify that background noise is generated. |

## Verifying Your Configuration

You can check the validity of your voice-port configuration by performing the following tasks:

- Pick up the handset of an attached telephony device and listen for a dial tone.
- Check for DTMF detection if you have a dial tone. If the dial tone stops when you dial a digit, the voice port is configured properly.
- Use the **show voice-port** command to verify that the data configured is correct.

## Troubleshooting Tips

If you are having trouble connecting a call and you suspect the problem is associated with the voice-port configuration, you can try to resolve the problem by performing the following tasks:

- Ping the associated IP address to confirm connectivity. If you cannot ping your destination, refer to the *Network Protocols Configuration Guide, Part 1* for Cisco IOS Release 12.1T.
- Use the **show voice-port command** to make sure that the port is enabled. If the port is offline, use the **no shutdown** command.
- If you have configured E&M interfaces, make sure that the values pertaining to your specific PBX setup, such as timing and type, are correct.
- Make sure the VICs are correctly installed. For more information, refer to the *Cisco WAN Interface Cards Hardware Installation Guide*.

# Fine-Tune E&M Voice Ports

In most cases, the default values for voice-port tuning commands are sufficient. Depending on the specifics of your particular network, you might need to adjust voice parameters involving timing, input gain, and output attenuation for E&M voice ports. Collectively, these commands are referred to as voice-port tuning commands.

If you need to change the default tuning configuration for E&M voice ports, use the following commands, beginning in privileged EXEC mode:

| | Command | Task | Valid Entries | Default Values |
|---|---|---|---|---|
| **Step 1** | **configure terminal** | Enter the global configuration mode. | | |
| **Step 2** | **voice-port** *slot-number/port* | Identify the voice port you want to configure, and enter the voice port configuration mode. | | |
| **Step 3** | **input gain** *value* | Specify (in dB) the amount of gain to be inserted at the receiver side of the interface. | –6 to 14 dB | 0 dB |
| **Step 4** | **output attenuation** *value* | Specify (in dB) the amount of attenuation at the transmit side of the interface. | 0 to 14 dB | 0 dB |
| **Step 5** | **echo-cancel enable** | Enable echo-cancellation of voice that is sent out of the interface and received back on the same interface. | | |
| **Step 6** | **echo-cancel coverage** *value* | Adjust the size (in milliseconds) of the echo-cancel. | 8, 16, 24, and 32 ms | 16 ms |

| | Command | Task | Valid Entries | Default Values |
|---|---|---|---|---|
| Step 7 | **non-linear** | Enable nonlinear processing, which shuts off any signal if no near-end speech is detected. (Nonlinear processing is used with echo-cancellation.) | | |
| Step 8 | **timeouts initial** *seconds* | Specify the number of seconds the system will wait for the caller to input the first digit of the dialed digits. | 0 to 120 sec | 10 sec |
| Step 9 | **timeouts interdigit** *seconds* | Specify the number of seconds the system will wait (after the caller has input the initial digit) for the caller to input a subsequent digit. | 0 to 120 sec | 10 sec |
| Step 10 | | **Specify timing parameters for each of these commands.** | | |
| | **timing clear-wait** *milliseconds* | | 200 to 2000 ms | |
| | **timing delay-duration** *milliseconds* | | 100 to 5000 ms<br>20 to 2000 ms | |
| | **timing delay-start** *milliseconds* | | 0 to 5000 ms | |
| | **timing dial-pulse min-delay** *milliseconds* | | 50 to 100 ms | |
| | **timing digit** *milliseconds* | | 50 to 500 ms | |
| | **timing inter-digit** *milliseconds* | | 10 to 20 pps | |
| | **timing pulse** *pulses-per-second* | | 100 to 1000 ms | |
| | **timing pulse-inter-digit** *milliseconds* | | 100 to 400 ms | |
| | **timing wink-duration** *milliseconds* | | 100 to 5000 ms | |
| | **timing wink-wait** *milliseconds* | | | |

**Note** After you change any voice-port command, we recommend that you cycle the port by using the **shutdown** and **no shutdown** commands.

# Additional VoIP Dial Peer Configurations

Depending on how you have configured your network interfaces, you might need to configure additional VoIP dial-peer parameters This section describes the following topics:

- Configure IP Precedence for Dial Peers
- Configure RSVP for Dial Peers
- Configure CODEC and VAD for Dial Peers

## Configure IP Precedence for Dial Peers

Use the **ip precedence** command to give voice packets a higher priority than other IP data traffic. The **ip precedence** command should also be used if RSVP is not enabled and you would like to give voice packets a priority over other IP data traffic. IP precedence scales better than RSVP, but provides no admission control.

To give real-time voice traffic precedence over other IP network traffic, use the following global configuration commands:

| | Command | Task |
|---|---|---|
| Step 1 | **dial-peer voice** *number* **voip** | Enter the dial peer configuration mode to configure a VoIP dial peer. |
| Step 2 | **ip precedence** *number* | Select a precedence level for the voice traffic associated with that dial peer. |

In IP precedence, the numbers 1 through 5 identify classes for IP flows; the numbers 6 through 7 are used for network and backbone routing and updates.

For example, to ensure that voice traffic associated with VoIP dial peer 103 is given a higher priority than other IP network traffic, enter the following:

```
Router(config)# dial-peer voice 103 voip
Router(config-dial-peer)# ip precedence 5
```

In this example, when an IP call leg is associated with VoIP dial peer 103, all packets transmitted to the IP network via this dial peer will have their precedence bits set to 5. If the networks receiving these packets have been configured to recognize precedence bits, the packets are given priority over packets with a lower configured precedence value.

## Configure RSVP for Dial Peers

RSVP must be enabled at each LAN or WAN interface that voice packets will travel across. After enabling RSVP, you must use the **req-qos** dial-peer configuration command to request an RSVP session and configure the QoS for each VoIP dial peer. Otherwise, no bandwidth is reserved for voice traffic.

To configure controlled-load QoS for VoIP dial peer 108, enter the following global configuration commands:

```
Router(config)# Dial-peer voice 108 voip
Router(config-dial-peer)# req-qos controlled-load
Router(config-dial-peer)# session target ipv4:10.0.0.8
```

In this example, every time a connection is made through VoIP dial peer 108, an RSVP reservation request is made between the local router, all intermediate routers in the path, and the final destination router.

**Note** We recommend that you select **controlled-load** for the requested QoS. The controlled-load service uses admission (or capacity) control to ensure that preferential service is received even when the bandwidth is overloaded.

To generate a Simple Network Management Protocol (SNMP), use the following commands beginning in global configuration mode:

| | Command | Task |
|---|---|---|
| Step 1 | **dial-peer voice** *number* **voip** | Enter the dial peer configuration mode to configure a VoIP dial peer. |
| Step 2 | **acc-qos** [**best-effort** \| **controlled-load** \| **guaranteed-delay**] | Generate an SNMP event if the QoS for a dial peer drops below a specified level. |

**Note** RSVP reservations are one-way only. If you configure RSVP, the VoIP dial peers on either side of the connection must be configured for RSVP.

# Configure CODEC and VAD for Dial Peers

CODEC typically is used to transform analog signals into a digital bit stream and digital signals back into analog signals—in this case, it specifies the voice coder rate of speech for a dial peer. Voice activity detection (VAD) is used to disable the transmission of silence packets. CODEC and VAD values for a dial peer determine how much bandwidth the voice session uses.

## Configure CODEC for a VoIP Dial Peer

To specify a voice coder rate for a selected VoIP dial peer, use the following commands, beginning in global configuration mode:

| | Command | Task |
|---|---|---|
| Step 1 | **dial-peer voice** *number* **voip** | Enter the dial peer configuration mode to configure a VoIP dial peer. |
| Step 2 | **codec** [**g711alaw** \| **g711ulaw** \| **g729r8** \| **g729r8** \| ...] | Specify the desired voice coder rate of speech. |

The default for the **codec** command is **g729r8**; normally, the default configuration for this command is the most desirable. However, if you are operating on a high bandwidth network and voice quality is of the highest importance, you should configure the **codec** command for **g711alaw** or **ulaw**. Using this value results in better voice quality, but it also requires higher bandwidth requirements for voice.

For example, to specify a CODEC rate of **g711alaw** for VoIP dial peer 108, enter the following:

```
Router(config)# dial-peer voice 108 voip
Router(config-dial-peer)# codec g711alaw
```

## Configure VAD for a VoIP Dial Peer

To disable the transmission of silence packets and enable VAD for a selected VoIP dial peer, use the following global configuration commands:

| | Command | Task |
|---|---|---|
| Step 1 | **dial-peer voice** *number* **voip** | Enter the dial peer configuration mode to configure a VoIP dial peer. |
| Step 2 | **vad** | Disable the transmission of silence packets . |

The default for the **vad** command is enabled; normally, the default configuration for this command is the most desirable. If you are operating on a high bandwidth network and voice quality is of the highest importance, you should disable VAD. Using this value results in better voice quality, but it also requires higher bandwidth requirements for voice.

For example, to enable VAD for VoIP dial peer 108, enter the following:

```
Router(config)# Dial-peer voice 108 voip
Router(config-dial-peer)# vad
```

# Configure Frame Relay for VoIP

You need to take certain factors into consideration when configuring VoIP so that it runs smoothly over Frame Relay. A public Frame Relay cloud provides no guarantees for QoS. For real-time traffic to be transmitted in a timely manner, the data rate must not exceed the committed information rate (CIR), or there is the possibility that packets are dropped. In addition, Frame Relay traffic shaping and RSVP are mutually exclusive. This is particularly important to remember if multiple data link connection identifiers (DLCIs) are carried on a single interface.

For Frame Relay links with slow output rates (less than or equal to 64 kbps), where data and voice are being transmitted over the same permanent virtual circuit (PVC), we recommend the following solutions:

- Separate DLCIs for voice and data—By providing a separate subinterface for voice and data, you can use the appropriate QoS tool per line. For example, each DLCI would use 32 kbps of a 64-kbps line.
  - Apply adaptive traffic shaping to both DLCIs.
  - Use RSVP or IP precedence to prioritize voice traffic.
  - Use compressed RTP to minimize voice packet size.
  - Use weighted fair queuing to manage voice traffic.
- Lower maximum transmission unit (MTU) size—Voice packets are generally small. By lowering the MTU size (for example, to 300 bytes), large data packets can be broken up into smaller data packets that can more easily be interwoven with voice packets.

**Note** Lowering the MTU size affects data throughput speed.

- CIR equal to line rate—Make sure that the data rate does not exceed the CIR. This is accomplished through generic traffic shaping.

  - Use RSVP or IP precedence to prioritize voice traffic.
  - Use compressed RTP to minimize voice packet header size.
- Traffic shaping—Use adaptive traffic shaping to slow the output rate based on the backward explicit congestion notification (BECN). If the feedback from the switch is ignored, packets (both data and voice) might be discarded. Because the Frame Relay switch does not distinguish between voice and data packets, voice packets could be discarded, which would result in a deterioration of voice quality.
  - Use RSVP, compressed RTP, reduced MTU size, and adaptive traffic shaping based on BECN to hold data rate to CIR.
  - Use generic traffic shaping to obtain a low interpacket wait time. For example, set committed burst (Bc) to 4000 to obtain an interpacket wait of 125 milliseconds.

In Cisco IOS Release 12.1T, Frame Relay traffic shaping is not compatible with RSVP. We suggest one of the following workarounds:

- Provision the Frame Relay PVC to have the CIR equal to the port speed.
- Use generic traffic shaping with RSVP.

# Frame Relay for VoIP Configuration Example

For Frame Relay, it is customary to configure a main interface and several subinterfaces with one subinterface per PVC. The following example configures a Frame Relay main interface and a subinterface so that voice and data traffic can be successfully transported:

```
interface Serial0/0
 mtu 300
 no ip address
 encapsulation frame-relay
 no ip route-cache
 no ip mroute-cache
 fair-queue 64 256 1000
 frame-relay ip rtp header-compression

interface Serial1/0 point-to-point
 mtu 300
 ip address 40.0.0.7 255.0.0.0
 ip rsvp bandwidth 48 48
 no ip route-cache
 no ip mroute-cache
 bandwidth 64
 traffic-shape rate 32000 4000 4000
 frame-relay interface-dlci 16
 frame-relay ip rtp header-compression
```

In this configuration example, the main interface is configured as follows:

- MTU size is 300 bytes.
- No IP address is associated with this serial interface. The IP address must be assigned for the subinterface.
- Encapsulation method is Frame Relay.
- Fair-queuing is enabled.
- IP RTP header compression is enabled.

The subinterface is configured as follows:

- MTU size is inherited from the main interface.
- IP address for the subinterface is specified.
- RSVP is enabled to use the default value, which is 75 percent of the configured bandwidth.
- Bandwidth is set to 64 kbps.
- Generic traffic shaping is enabled with 32-kbps CIR where committed burst (Bc) = 4000 bits and excess burst (Be) = 4000 bits.
- Frame Relay DLCI number is specified.
- IP RTP header compression is enabled.

**Note** When traffic bursts over the CIR, the output rate is held at the speed configured for the CIR (for example, traffic will not go beyond 32 kbps if CIR is set to 32 kbps).

For more information about configuring Frame Relay for VoIP, refer to the "Configuring Frame Relay" chapter in the *Wide-Area Networking Configuration Guide* for Cisco IOS Release 12.1T.

# Configure Microsoft NetMeeting for VoIP

VoIP can be used with Microsoft NetMeeting (Version 2.x) when your router is used as the voice gateway. Use the latest version of DirectX drivers from Microsoft on your PC to improve the voice quality of NetMeeting.

## Configure VoIP to Support Microsoft NetMeeting

To configure VoIP to support NetMeeting, create a VoIP dial peer that has the following information:

- Session Target—IP address or domain name system (DNS) name of the PC running NetMeeting
- CODEC—**g711ulaw** or **g711alaw**

## Configure Microsoft NetMeeting for VoIP

To configure NetMeeting to work with VoIP, complete the following steps:

**Step 1** From the Tools menu in the NetMeeting application, select **Options**. NetMeeting will display the Options dialog box.

**Step 2** Click the **Audio** tab.

**Step 3** Select the "Calling a telephone using NetMeeting" check box.

**Step 4** Enter the IP address of your router in the **IP address** field.

**Step 5** Under **General**, click **Advanced**.

**Step 6** Select the "Manually configured compression settings" check box.

**Step 7** Select the CODEC value **CCITT ulaw 8000Hz**.

**Step 8** Click the **Up** button until this CODEC value is at the top of the list.

**Step 9** Click **OK** to exit.

## Initiate a Call Using Microsoft NetMeeting

To initiate a call using Microsoft NetMeeting, perform the following steps:

**Step 1** Click the **Call** icon from the NetMeeting application. Microsoft NetMeeting opens the call dialog box.

**Step 2** From the Call dialog box, select **call using H.323 gateway**.

**Step 3** Enter the telephone number in the **Address** field. (Enter 1 and the area code followed by the seven-digit telephone number in the following format 1Nxx-Nxx-xxxx, with N = digits 2 through 9 and x = digits 0 through 9.)

**Step 4** Click **Call** to initiate a call to your router from Microsoft NetMeeting.

# 3

# VoIP Configuration Examples

This chapter demonstrates how to configure VoIP in four different scenarios. The actual VoIP configuration procedure depends on the actual topology of your voice network. The following configuration examples should give you a starting point. These configuration examples would need to be customized to reflect your network topology.

Configuration procedures are supplied for the following scenarios:

- FXS-to-FXS Connection Using RSVP
- Linking PBX Users with E&M Trunk Lines
- FXO Gateway to PSTN
- FXO Gateway to PSTN (PLAR Mode)

## FXS-to-FXS Connection Using RSVP

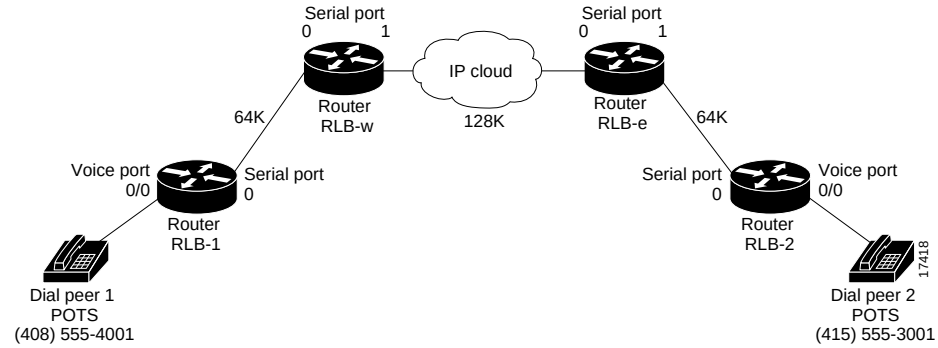
The following example shows how to configure VoIP for simple FXS-to-FXS connection.

In this example, a very small company with two offices decides to integrate VoIP in its existing IP network. One basic telephony device is connected to Router RLB-1; therefore, Router RLB-1 is configured for one POTS dial peer and one VoIP dial peer. Router RLB-w and Router RLB-e establish the WAN connection between the two offices. Because one POTS telephony device is connected to Router RLB-2, it is also configured for one POTS dial peer and one VoIP dial peer.

In this example, only the calling end (Router RLB-1) is requesting RSVP.

Figure 1 illustrates the topology of this FXS-to-FXS connection example.

***Figure 1 FXS-to-FXS Connection Example***




## Configuration for Router RLB-1

```
hostname RLB-1

! Create voip dial-peer 2
dial-peer voice 2 voip

! Define its associated telephone number and IP address
 destination-pattern 14155553001
 sess-target ipv4:40.0.0.1

! Request RSVP
 req-qos controlled-load

! Create pots dial-peer 1
dial-peer voice 1 pots

! Define its associated telephone number and voice port
 destination-pattern 14085554001
 port 0/0

! Configure serial interface 0
interface serial1/0
 ip address 10.0.0.1 255.0.0.0
 no ip mroute-cache

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 48 48
 fair-queue 64 256 36
 clockrate 64000

router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

## Configuration for Router RLB-w

```
hostname RLB-w

! Configure serial interface 0
interface serial0/0
 ip address 10.0.0.2 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3

! Configure serial interface 1
interface serial1/0
 ip address 20.0.0.1 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3

! Configure IGRP
router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

## Configuration for Router RLB-e

```
hostname RLB-e

! Configure serial interface 0
interface serial0/0
 ip address 40.0.0.2 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3

! Configure serial interface 1
interface serial1/0
 ip address 20.0.0.2 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3
 clockrate 128000

! Configure IGRP
router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

### Configuration for Router RLB-2

```
hostname RLB-2

! Create pots dial-peer 2
dial-peer voice 2 pots

! Define its associated telephone number and voice-port
 destination-pattern 14155553001
 port 0/0

! Create voip dial-peer 1
dial-peer voice 1 voip

!Define its associated telephone number and IP address
 destination-pattern 14085554001
 sess-target ipv4:10.0.0.1

! Configure serial interface 0
interface serial1/0
 ip address 40.0.0.1 255.0.0.0
 no ip mroute-cache

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3
 clockrate 64000

! Configure IGRP
router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

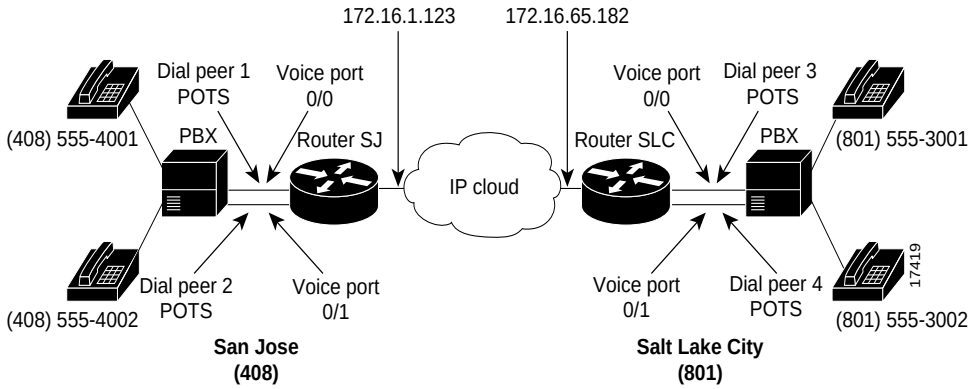
## Linking PBX Users with E&M Trunk Lines

The following example shows how to configure VoIP to link PBX users with E&M trunk lines.

In this example, a company decides to connect two offices: one in San Jose, California, and the other in Salt Lake City, Utah. Each office has an internal telephone network using PBX, connected to the voice network by an E&M interface. Both the Salt Lake City and the San Jose offices are using E&M Port Type II, with four-wire operation and ImmediateStart signaling. Each E&M interface connects to the router using two voice interface connections. Users in San Jose dial *801-555* and then the extension number to reach a destination in Salt Lake City. Users in Salt Lake City dial *408-555* and then the extension number to reach a destination in San Jose.

Figure 2 illustrates the topology of this connection example.

*Figure 2 Linking PBX Users with E&M Trunk Lines Example*




**Note** This example assumes that the company has already established a working IP connection between its two remote offices.

## Router SJ Configuration

```
hostname router SJ

!Configure pots dial-peer 1
dial-peer voice 1 pots
 destination-pattern 1408555....
 port 0/0

!Configure pots dial-peer 2
dial-peer voice 2 pots
 destination-pattern 1408555....
 port 0/1

!Configure voip dial-peer 3
dial-peer voice 3 voip
 destination-pattern 1801555....
 session target ipv4:172.16.65.182
 ip precedence 5

!Configure the E&M interface
voice-port 0/0
 signal immediate
 operation 4-wire
 type 2

voice-port 0/1
 signal immediate
 operation 4-wire
 type 2

!Configure the serial interface 0
interface serial1/0
 ip address 172.16.1.123 255.255.0.0
 no shutdown
```

### Router SLC Configuration

```
hostname router SLC

!Configure pots dial-peer 3
dial-peer voice 3 pots
 destination-pattern 1801555....
 port 0/0

!Configure pots dial-peer 4
dial-peer voice 4 pots
 destination-pattern 1801555....
 port 0/1

!Configure voip dial-peer 1
dial-peer voice 1 voip
 destination-pattern 1408555....
 session target ipv4:172.16.1.123
 ip precedence 5

!Configure the E&M interface
voice-port 0/0
 signal immediate
 operation 4-wire
 type 2

voice-port 0/1
 signal immediate
 operation 4-wire
 type 2

!Configure the serial interface 0
interface serial1/0
 ip address 172.16.65.182 255.255.0.0
 no shutdown
```

**Note** PBXs should be configured to pass all DTMF signals to the router. We recommend that you do not configure, store, and forward tone.

**Note** If you change the gain or the telephony port, make sure that the telephony port still accepts DTMF signals.

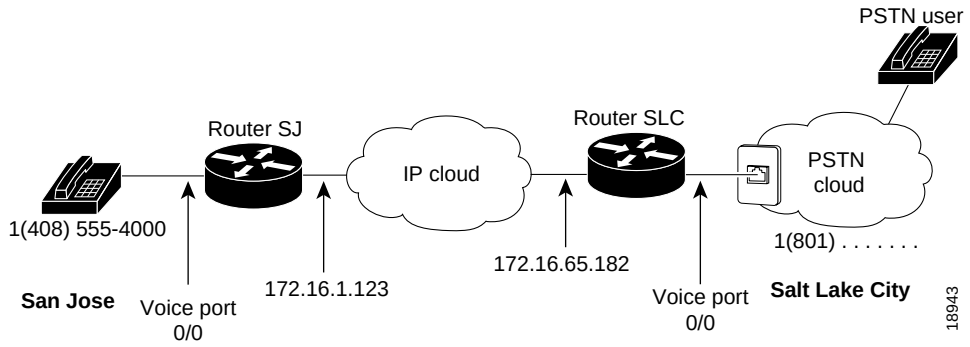
## FXO Gateway to PSTN

FXO interfaces provide a gateway from the VoIP network to the analog PSTN or to a PBX that does not support E&M signaling so that users can reach telephones and fax machines outside the VoIP network.

In this example, users connected to Router SJ in San Jose, California, can reach PSTN users in Salt Lake City, Utah, via Router SLC. Router SLC in Salt Lake City is connected directly to the PSTN through an FXO interface.

Figure 3 illustrates the topology of this connection example.

*Figure 3 FXO Gateway to PSTN Example*




**Note** This example assumes that the company has already established a working IP connection between its two remote offices.

## Router SJ Configuration

```
hostname router SJ

! Configure pots dial-peer 1
dial-peer voice 1 pots
 destination-pattern 14085554000
 port 0/0

! Configure voip dial-peer 2
dial-peer voice 2 voip
 destination-pattern 1801.......
 session target ipv4:172.16.65.182
 ip precedence 5

! Configure serial interface 0
interface serial1/0
 clock rate 2000000
 ip address 172.16.1.123 255.255.0.0
 no shutdown
```

## Router SLC Configuration

```
hostname router SLC

! Configure pots dial-peer 1
dial-peer voice 1 pots
 destination-pattern 1801.......
 port 0/0

! Configure voip dial-peer 2
dial-peer voice 2 voip
 destination-pattern 14085554000
 session target ipv4:172.16.1.123
 ip precedence 5

! Configure serial interface 0
interface serial1/0
 ip address 172.16.65.182 255.255.0.0
 no shutdown
```

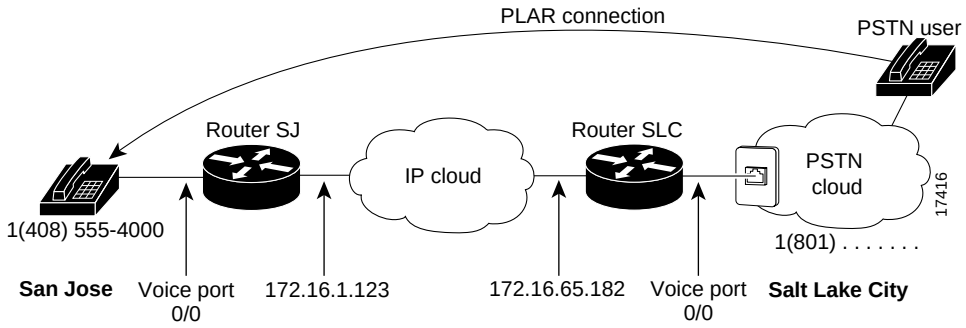
# FXO Gateway to PSTN (PLAR Mode)

The following example shows an FXO gateway to PSTN connection in PLAR mode.

In this example, PSTN users in Salt Lake City, Utah, can dial a local number and establish a private line connection in a remote location. As in the previous example, Router SLC in Salt Lake City is connected directly to the PSTN through an FXO interface.

Figure 4 illustrates the topology of this connection example.

***Figure 4 FXO Gateway to PSTN (PLAR Mode) Example***




**Note** This example assumes that the company has already established a working IP connection between its two remote offices.

## Router SJ Configuration

```
hostname router SJ

! Configure pots dial-peer 1
dial-peer voice 1 pots
 destination-pattern 14085554000
 port 0/0

! Configure voip dial-peer 2
dial-peer voice 2 voip
 destination-pattern 1801.......
 session target ipv4:172.16.65.182
 ip precedence 5

! Configure the serial interface 0
interface serial1/0
 clock rate 2000000
 ip address 172.16.1.123 255.255.0.0
 no shutdown
```

## Router SLC Configuration

```
hostname router SLC

! Configure pots dial-peer 1
dial-peer voice 1 pots
 destination-pattern 1801.......
 port 0/0

! Configure voip dial-peer 2
dial-peer voice 2 voip
 destination-pattern 14085554000
 session target ipv4:172.16.1.123
 ip precedence 5

! Configure the voice port
voice port 0/0
connection plar 14085554000

! Configure the serial interface 0
interface serial1/0
 ip address 172.16.65.182 255.255.0.0
 no shutdown
```

# 4

# VoIP Commands

This chapter provides an alphabetical listing of all of the VoIP commands that are new or specific to the Cisco 1751 router. All other commands used with this feature are documented in the Cisco IOS Release 12.1T command reference documents.

Table 1 lists and describes the commands in this chapter that are used to configure and monitor VoIP.

***Table 1 Commands Used to Configure and Monitor VoIP***

| Command | Description |
|---|---|
| **acc-qos** | Generate an SNMP event if the QoS drops below a specified level. |
| **answer-address** | Specify the full E.164 telephone number to identify the dial peer of an incoming call. |
| **codec** | Specify the voice coder rate of speech for a dial peer. |
| **comfort-noise** | Specify whether or not background noise should be generated. |
| **connection** | Specify a connection mode for a specified voice port. |
| **cptone** | Configure a voice call progress tone locale. |
| **description** | Include a description of what this voice port is connected to. |
| **destination-pattern** | Specify either the prefix or the full E.164 telephone number to be used for a dial peer. |
| **dial-control-mib** | Specify attributes for the call history table. |
| **dial-peer voice** | Enter the dial peer configuration mode. |
| **dial-type** | Specify the type of out-dialing for voice-port interfaces. |
| **echo-cancel coverage** | Adjust the size of the echo cancel. |
| **echo-cancel enable** | Enable the echo cancel feature. |
| **expect-factor** | Specify when the router will generate an alarm to the network manager. |
| **fax-rate** | Establish the rate at which a fax is sent to the specified dial peer. |
| **icpif** | Specify the Calculated Planning Impairment Factor (CPIF) for calls sent by a dial peer. |
| **impedance** | Specify the terminating impedance of a voice-port interface. |
| **input gain** | Configure a specific input gain value. |
| **ip precedence** | Set IP precedence (priority) for packets sent by the dial peer. |

*Table 1 Commands Used to Configure and Monitor VoIP*

| Command | Description |
|---|---|
| **ip udp checksum** | Calculate the UDP checksum for voice packets transmitted by the dial peer. |
| **music-threshold** | Specify the threshold for on-hold music for a specified voice port. |
| **non-linear** | Enable nonlinear processing in the echo canceller. |
| **num-exp** | Define how to expand an extension number into a particular destination pattern. |
| **operation** | Select a specific cabling scheme for E&M ports. |
| **output attenuation** | Configure a specific output attenuation value. |
| **port** | Associate a dial peer with a specific voice port. |
| **prefix** | Specify the prefix of the dialed digits for this dial peer. |
| **req-qos** | Specify the desired QoS to be used in reaching a specified dial peer. |
| **ring frequency** | Specify the ring frequency for a specified FXS voice port. |
| **ring number** | Specify the number of rings for a specified FXO voice port. |
| **session protocol** | Establish a session protocol for calls between the local and remote routers . |
| **session target** | Specify a network-specific address for a specified dial peer. |
| **show call active voice** | Show the active call table. |
| **show call history voice** | Display the call-history table. |
| **show controllers voice** | Display information about voice related hardware. |
| **show diag** | Display hardware information for the router. |
| **show dial-peer voice** | Display configuration information for dial peers. |
| **show dialplan incall number** | Pair different voice ports and telephone numbers together for troubleshooting. |
| **show dialplan number** | Show which dial peer is reached when a particular telephone number is dialed. |
| **show num-exp** | Show the number expansions configured. |
| **show voice dsp** | Display current status of all DSP voice channels |
| **show voice port** | Display configuration information about a specific voice port. |
| **shutdown (dial-peer configuration)** | Change the administrative state of the selected dial peer from up to down. |
| **shutdown (voice-port configuration)** | Take the voice ports for a specific VIC offline. |
| **signal** | Specify the type of signaling for a voice port. |
| **snmp enable peer-trap poor-qov** | Generate poor-quality-of-voice notification for applicable calls associated with VoIP dial peers. |
| **snmp-server enable traps** | Enable the router to send SNMP traps. |
| **snmp trap link-status** | Enable SNMP trap messages to be generated when this voice port is brought up or down. |
| **timeouts initial** | Configure the initial digit timeout value for a specified voice port. |
| **timeouts interdigit** | Configure the interdigit timeout value for a specified voice port. |
| **timing** | Specify timing parameters for a specified voice port. |
| **type** | Specify the E&M interface type. |

*Table 1 Commands Used to Configure and Monitor VoIP*

| Command | Description |
|---|---|
| **vad** | Enable VAD for the calls using this dial peer. |
| **voice-port** | Enter the voice port configuration mode. |

A subset of the commands listed are voice-port commands. Different voice signaling types support different voice-port commands. Table 2 lists the router voice-port commands and the signaling types supported.

*Table 2 Router Voice-Port Commands and Signaling Types Supported*

| Voice-Port Command | FXO | FXS | E&M |
|---|---|---|---|
| comfort-noise | – | – | – |
| connection | – | – | – |
| cptone | X | X | X |
| description | X | X | X |
| dial-type | X | – | X |
| echo-cancel coverage | – | – | – |
| echo-cancel enable | – | – | – |
| impedance | X | X | X |
| input gain | X | X | X |
| music-threshold | – | – | – |
| non-linear | – | – | – |
| operation | – | – | X |
| output attenuation | X | X | X |
| ring frequency | – | X | – |
| ring number | X | – | – |
| shutdown | X | X | X |
| signal | X | X | X |
| snmp trap link-status | – | – | – |
| timeouts initial | – | – | – |
| timeouts interdigit | – | – | – |
| timing | – | – | – |
| timing keywords: | – | – | – |
| clear-wait | – | – | X |
| delay-duration | – | – | X |
| delay-start | – | – | X |
| delay-with-integrity | – | – | X |
| digit | X | X | X |
| inter-digit | X | X | X |

*Table 2 Router Voice-Port Commands and Signaling Types Supported (Continued)*

| Voice-Port Command | FXO | FXS | E&M |
|---|---|---|---|
| pulse | X | – | X |
| pulse-inter-digit | X | – | X |
| wink-duration | – | – | X |
| wink-wait | – | – | X |
| type | – | – | X |

# acc-qos

To generate an SNMP event if the QoS for a dial peer drops below a specified level, use the **acc-qos** dial-peer configuration command. Use the **no** form of this command to use the default value for this feature.

**acc-qos** {**best-effort** | **controlled-load** | **guaranteed-delay**}
**no acc-qos**

## Syntax Description

| | |
|---|---|
| **best-effort** | RSVP makes no bandwidth reservation. |
| **controlled-load** | RSVP guarantees a single level of preferential service, presumed to correlate to a delay boundary. The controlled load service uses admission (or capacity) control to assure that preferential service is received even when the bandwidth is overloaded. |
| **guaranteed-delay** | RSVP reserves bandwidth and guarantees a minimum bit rate and preferential queuing if the bandwidth reserved is not exceeded. |

## Command Modes

Dial-peer configuration.

## Usage Guidelines

Use the **acc-qos** dial-peer command to generate an SNMP event if the QoS for specified dial peer drops below the specified level. When a dial peer is used, the Cisco IOS software reserves a certain amount of bandwidth so that the selected QoS can be provided. Cisco IOS software uses RSVP to request QoS guarantees from the network.

To select the most appropriate value for this command, you need to be familiar with the amount of traffic this connection supports and what kind of impact you are willing to have on it. The Cisco IOS software generates a trap message when the bandwidth required to provide the selected QoS is not available.

This command only applies to VoIP peers.

### Example

The following example selects guaranteed-delay as the specified level below which an SNMP trap message is generated:

```
dial-peer voice 10 voip
 acc-qos guaranteed-delay
```

### Related Commands

**req-qos**

# answer-address

To specify the full E.164 telephone number to be used to identify the dial peer of an incoming call, use the **answer-address** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**answer-address** [+]*string*
**no answer-address**

### Syntax Description

| | |
|---|---|
| *string* | Series of digits that specify the E.164 or private dialing plan telephone number:<br>• Digits 0 through 9, letters A through D, pound sign (#), and asterisk (*), which represent specific digits that can be entered.<br>• Plus sign (+), which is optionally used as the first digit to indicate an E.164 standard number.<br>• Comma (,), which inserts a pause between digits.<br>• Period (.), which is used as a wild-card character and matches any entered digit. |

### Default

Enabled with a null string.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **answer-address** command to identify the origin (or dial peer) of incoming calls from the IP network. Cisco IOS software identifies the dial peers of a call in one of two ways: either by identifying the interface through which the call is received or through the telephone number configured with the **answer-address** command. In the absence of a configured telephone number, the dial peer associated with the interface is associated with the incoming call.

For calls coming in from a POTS interface, the **answer-address** command is not used to select an incoming dial peer. The incoming POTS dial peer is selected on the basis of the port configured for that dial peer.

This command applies to both VoIP and POTS dial peers.

**Note** The Cisco IOS software does not check the validity of the E.164 telephone number; it accepts any series of digits as a valid number.

### Example

The following example configures the E.164 telephone number, *14085559626*, as the dial peer of an incoming call:

```
dial-peer voice 10 pots
 answer-address 14085559626
```

### Related Commands

**destination-pattern**
**port**
**prefix**

# codec

To specify the voice coder rate of speech for a dial peer, use the **codec** dial-peer configuration command. Use the **no** form of this command to reset the default value for this command.

**codec** {**g711alaw** | **g711ulaw** | **g723ar53** | **g723ar63** | **g723r53** | **g723r63** | **g726r16** | **g726r24** | **g726r32** | **g729br8** | **g729r8**}
**no codec**

### Syntax Description

| | |
|---|---|
| **g711alaw** | G.711 A-Law 64,000 bits per second (bps). |
| **g711ulaw** | G.711 U-Law 64,000 bps. |
| **g723ar53** | G.723.1 ANNEX-A 5,300 bps. |
| **g723ar63** | G.723.1 ANNEX-A 6,300 bps. |
| **g723r53** | G.723.1 5,300 bps. |
| **g723r63** | G.723.1 6,300 bps. |
| **g726r16** | G.726 16,000 bps. |
| **g726r24** | G.726 24,000 bps. |
| **g726r32** | G.726 32,000 bps. |
| **g729br8** | G.729 ANNEX-B 8,000 bps. |
| **g729r8** | G.729 8,000 bps. |

### Default

**g729r8**.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **codec** command to define a specific voice coder rate of speech for a dial peer.

For toll quality, use **g711alaw** or **g711ulaw**. These values provide high-quality voice transmission, but use a significant amount of bandwidth. For almost toll quality (and a significant savings in bandwidth), use the **g729r8** value.

If **codec**-command values for the VoIP peers of a connection do not match, the call fails.

This command only applies to VoIP peers.

**Note** Prior to Cisco IOS Release 12.0(5)T, **g729r8** is implemented in the pre-IETF format; thereafter it is implemented in the standard IETF format. Whenever new images, from Release 12.0(5)T or later, interoperate with older versions of VoIP (when the **g729r8** codec was not compliant with the IETF standard), users can hear garbled voices and ringback on either end of the connection. To avoid this problem, configure the dial peers with the **g729r8 pre-ietf** argument.

### Example

The following example configures a voice coder rate that provides toll quality and uses a relatively high amount of bandwidth:

```
dial-peer voice 10 voip
 codec g711alaw
```

# comfort-noise

To specify whether or not background noise should be generated, use the **comfort-noise** voice-port configuration command. Use the **no** form of this command to disable this feature.

**comfort-noise**
**no comfort-noise**

### Syntax Description

This command has no arguments or keywords.

### Default

Enabled.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **comfort-noise** command to generate background noise to fill silent gaps during calls if VAD is activated. If **comfort noise** is not enabled and VAD is enabled at the remote end of the connection, the user hears dead silence when the remote party is not speaking.

The configuration of **comfort noise** only affects the silence generated at the local interface; it does not affect the use of VAD on either end of the connection or the silence generated at the remote end of the connection.

### Example

The following example enables background noise:

```
voice port 0/0
 comfort-noise
```

### Related Commands

**vad**

# connection

To specify a connection mode for a specified voice port, use the **connection** voice-port configuration command. Use the **no** form of this command to disable the selected connection mode.

**connection** {**plar** | **trunk** } *string*
**no connection** {**plar** | **trunk** } *string*

## Syntax Description

| | |
|---|---|
| **plar** | Private line auto ringdown (PLAR) connection. PLAR connection associates a dial peer directly with an interface; when an interface goes off-hook, the dial peer sets up the second call leg and creates a conference call without the caller having to dial any digits. |
| trunk | Straight tie-line connection to a private branch exchange (PBX). |
| *string* | Destination telephone number. Valid entries are any series of digits that specify the E.164 telephone number. |

## Default

No connection.

## Command Mode

Voice-port configuration.

## Usage Guidelines

Use the **connection** command to specify a connection mode for a specific interface. Use the **connection plar** command to specify a PLAR interface. The string you configure for this command is used as the called number for all calls coming in over this voice port. The destination dial peer is determined on the basis of this called number.

Use the **connection trunk** command to specify a straight tie-line connection to a PBX. This command can be used for E&M-to-E&M trunks, FXO-to-FXS trunks, and FXS-to-FXS trunks. Signaling is transported for E&M-to-E&M trunks and FXO-to-FXS trunks; signaling will not be transported for FXS-to-FXS trunks.

If the **connection** command is not configured, the standard session application creates a dial tone when the interface goes off-hook until enough digits are collected to match a dial peer and complete the call.

## Example

The following example selects **plar** as the connection mode and a destination telephone number of *14085559262*:

```
voice port 0/0
 connection plar 14085559262
```

The following example selects **trunk** as the connection mode and a destination telephone number of *14085559262*:

```
voice port 0/0
 connection trunk 14085559262
```

## Related Commands

**session protocol**

# cptone

To configure a voice call progress tone locale, use the **cptone** voice-port configuration command. Use the **no** form of this command to disable this feature.

**cptone** {**australia** | **brazil** | **china** | **finland** | **france** | **germany** | **japan** | **northamerica** | **unitedkingdom**}
**no cptone**

## Syntax Description

| | |
|---|---|
| **australia** | Analog voice interface-related default tone, ring, and cadence setting for Australia. |
| **brazil** | Analog voice interface-related default tone, ring, and cadence setting for Brazil. |
| **china** | Analog voice interface-related default tone, ring, and cadence setting for China. |
| **finland** | Analog voice interface-related default tone, ring, and cadence setting for Finland. |
| **france** | Analog voice interface-related default tone, ring, and cadence setting for France. |
| **germany** | Analog voice interface-related default tone, ring, and cadence setting for Germany. |
| **japan** | Analog voice interface-related default tone, ring, and cadence setting for Japan. |
| **northamerica** | Analog voice interface-related default tone, ring, and cadence setting for North America. |
| **unitedkingdom** | Analog voice interface-related default tone, ring, and cadence setting for the United Kingdom. |

## Default

**northamerica**.

## Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **cptone** command to specify a regional analog voice interface-related tone, ring, and cadence setting for a specified voice port. This command only affects the tones generated at the local interface. It does not affect any information passed to the remote end of a connection or any tones generated at the remote end of a connection.

### Example

The following example configures North America as the call progress tone locale:

```
voice port 0/0
 cptone northamerica
```

# description

To include a description of what this voice port is connected to, use the **description** voice-port configuration command. Use the **no** form of this command to disable this feature.

**description** *string*
**no description**

### Syntax Description

| | |
|---|---|
| *string* | Character string from 1 to 255 characters. |

### Default

Enabled with a null string.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **description** command to include descriptive text about this voice-port connection. This information is displayed when you issue a **show** command and does not affect the operation of the interface in any way.

### Example

The following example identifies this voice port as a connection to the purchasing department:

```
voice port 0/0
 description purchasing_dept
```

# destination-pattern

To specify either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer, use the **destination-pattern** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**destination-pattern** [+]*string*
**no destination-pattern**

## Syntax Description

| | |
|---|---|
| *string* | Series of digits that specify the E.164 or private dialing plan telephone number:<br>• Digits 0 through 9, letters A through D, pound sign (#), and asterisk (*), which represent specific digits that can be entered.<br>• Plus sign (+), which is optionally used as the first digit to indicate an E.164 standard number.<br>• Comma (,), which inserts a pause between digits.<br>• Period (.), which is used as a wild-card character and matches any entered digit. |

## Default

Enabled with a null string.

## Command Mode

Dial-peer configuration.

## Usage Guidelines

Use the **destination-pattern** command to define the E.164 telephone number for this dial peer. This pattern is used to match dialed digits to a dial peer. The dial peer is then used to complete the call.

This command applies to both VoIP and POTS dial peers.

**Note** The Cisco IOS software does not check the validity of the E.164 telephone number; it accepts any series of digits as a valid number.

## Example

The following example configures the E.164 telephone number, *14085557922*, for a dial peer:

```
dial-peer voice 10 pots
 destination-pattern 14085557922
```

## Related Commands

**answer-address**
**prefix**

# dial-control-mib

To specify attributes for the call history table, use the **dial-control-mib** global configuration command.

**dial-control-mib** {**max-size** *number* | **retain-timer** *number*}

### Syntax Description

| | |
|---|---|
| **max-size** *number* | Maximum size of the call history table. Valid entries are from 0 to 500 table entries. A value of 0 prevents any history from being retained. |
| **retain-timer** *number* | Length of time, in minutes, for entries in the call history table. Valid entries are from 0 to 2147483647 minutes. A value of 0 prevents any history from being retained. |

### Defaults

The default call history table length is 50 table entries. The default retain timer is 15 minutes.

### Command Mode

Global configuration.

### Usage Guidelines

The call history table contains a listing of all calls connected through the router in descending time order since VoIP was enabled. Use the **dial-control-mib** global configuration command to specify attributes for the call history table.

### Example

The following example configures the call history table to hold 400 entries, with each entry remaining in the table for 10 minutes:

```
configure terminal
 dial-control-mib max-size 400
 dial-control-mib retain-timer 10
```

# dial-peer voice

To enter the dial peer configuration mode (and specify the method of voice-related encapsulation), use the **dial-peer voice** global configuration command.

**dial-peer voice** *number* {**voip** | **pots**}

### Syntax Description

| | |
|---|---|
| *number* | Digit(s) defining a particular dial peer. Valid entries are from 1 to 2147483647. |
| **voip** | VoIP dial peer using voice encapsulation on the POTS network. |
| **pots** | POTS dial peer using VoIP encapsulation on the IP backbone. |

### Default

No dial peer configuration mode is preconfigured.

### Command Mode

Global configuration.

### Usage Guidelines

Use the **dial-peer voice** global configuration command to switch to the dial peer configuration mode from the global configuration mode. Use the **exit** command to exit the dial peer configuration mode and return to the global configuration mode.

### Example

The following example accesses the dial peer configuration mode and configures a POTS dial peer identified as dial peer 10:

```
configure terminal
 dial-peer voice 10 pots
```

### Related Commands

**voice-port**

# dial-type

To specify the type of out-dialing for voice-port interfaces, use the **dial-type** voice-port configuration command. Use the **no** form of this command to disable this feature.

**dial-type** {**dtmf** | **pulse**}
**no dial-type**

### Syntax Description

| | |
|---|---|
| **dtmf** | Touch-tone dialer. |
| **pulse** | Pulse dialer. |

### Default

**dtmf**.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **dial-type** command to specify an out-dialing type for an FXO or E&M voice-port interface; this command does not apply to FXS voice ports because they do not generate out-dialing. Voice ports can always detect DTMF and pulse signals. This command does not affect voice-port dialing detection.

The **dial-type** command affects out-dialing as configured for the dial peer.

### Example

The following example configures a voice port to support a touch-tone dialer:

```
voice port 0/0
 dial-type dtmf
```

# echo-cancel coverage

To adjust the size of the echo cancel, use the **echo-cancel coverage** voice-port configuration command. Use the **no** form of this command to reset this command to the default value.

**echo-cancel coverage** *value*
**no echo-cancel coverage** *value*

### Syntax Description

| | |
|---|---|
| *value* | Number of milliseconds (ms) the echo-canceller covers on a given signal. Valid values are 8, 16, 24, and 32 ms. |

### Default

16 ms.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **echo-cancel coverage** command to adjust the coverage size of the echo canceller. This command enables cancellation of voice that is sent out of the interface and received back on the same interface within the configured amount of time. If the local loop (the distance from the analog interface to the connected equipment producing the echo) is longer, the configured value of this command should be extended.

If you configure a longer value for this command, the echo canceller takes longer to converge; in this case, the user might hear a slight echo when the connection is initially set up. If the configured value for this command is too short, the user might hear some echo for the duration of the call because the echo canceller is not cancelling the longer delay echoes.

There is no echo or echo cancellation on the IP side of the connection.

**Note** This command is valid only if the echo cancel feature has been enabled. For more information, refer to the **echo-cancel enable** command.

### Example

The following example adjusts the size of the echo canceller to 16 ms:

```
voice port 0/0
 echo-cancel enable
 echo-cancel coverage 16
```

### Related Commands

**echo-cancel enable**

# echo-cancel enable

To enable the echo cancel feature, use the **echo-cancel enable** voice-port configuration command. Use the **no** form of this command to disable this feature.

**echo-cancel enable**
**no echo-cancel enable**

### Syntax Description

This command has no arguments or keywords.

### Default

Enabled for all interface types.

### Command Mode

Voice-port configuration.

### Usage Guidelines

The **echo-cancel** command enables cancellation of voice that is sent out of the interface and is received back on the same interface. Disabling echo cancellation might cause the remote side of a connection to hear an echo. Because echo cancellation is an invasive process that can minimally degrade voice quality, this command should be disabled if it is not needed.

The **echo-cancel** command does not affect the echo heard by the user on the analog side of the connection.

There is no echo path for a four-wire E&M interface. The echo canceller should be disabled for that interface type.

**Note** This command is valid only if the **echo-cancel coverage** command has been configured. For more information, refer to the **echo-cancel coverage** command.

### Example

The following example enables the echo cancel feature for 16-millisecond echo coverage:

```
voice port 0/0
 echo-cancel enable
 echo-cancel coverage 16
```

### Related Commands

**echo-cancel coverage**
**non-linear**

# expect-factor

To specify when the router generates an alarm to the network manager, indicating that the expected quality of voice has dropped, use the **expect-factor** dial-peer configuration command. Use the **no** form of this command to reset the default value for this command.

**expect-factor** *value*
**no expect-factor** *value*

### Syntax Description

| | |
|---|---|
| *value* | Integers that represent the ITU-T specification for quality of voice as described in G.113. Valid entries are from 0 to 20, with 0 representing toll quality. |

### Default

10.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

VoIP monitors the quality of voice received over the network. Use the **expect-factor** command to specify when the router generates an SNMP trap to the network manager.

This command only applies to VoIP peers.

### Example

The following example configures toll quality of voice when connecting to a dial peer:

```
dial-peer voice 10 voip
 expect-factor 0
```

# fax-rate

To establish the rate at which a fascimile (fax) is sent to the specified dial peer, use the **fax-rate** dial-peer configuration command. Use the **no** form of this command to reset the default value for this command.

**fax-rate**{**2400** | **4800** | **7200** | **9600** | **14400** | **disable** | **voice**}
**no fax-rate**

### Syntax Description

| | |
|---|---|
| **2400** | Fax transmission speed of 2400 bps. |
| **4800** | Fax transmission speed of 4800 bps. |
| **7200** | Fax transmission speed of 7200 bps. |
| 9600 | Fax transmission speed of 9600 bps. |
| 14400 | Fax transmission speed of 14,400 bps. |
| disable | Fax relay transmission capability disabled. |
| voice | Highest possible transmission speed allowed by voice rate. |

### Default

**voice**.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **fax-rate** command to specify the fax transmission rate to the specified dial peer.

The values for this command apply only to the fax transmission speed and do not affect the quality of the fax itself. The higher values provide a faster transmission speed but monopolize a significantly larger portion of the available bandwidth. Slower transmission speeds use less bandwidth.

If the **fax-rate** command is set above the **codec** command rate in the same dial peer, the data sent over the network for fax transmission exceeds the bandwidth reserved for RVSP. Because more network bandwidth is monopolized by the fax transmission, we do not recommend setting the **fax-rate** value higher than the **codec** command value. If the **fax-rate** value is set lower than the **codec-**command value, faxes take longer to transmit but use less bandwidth.

This command only applies to VoIP peers.

### Example

The following example configures a fax rate of 9600 bps for faxes sent to a dial peer:

```
dial-peer voice 10 voip
 fax-rate 9600
```

### Related Commands

**codec**

# icpif

To specify the Calculated Planning Impairment Factor (ICPIF) for calls sent by a dial peer, use the **icpif** dial-peer configuration command. Use the **no** form of this command to restore the default value for this command.

**icpif** *number*
**no icpif** *number*

### Syntax Description

| | |
|---|---|
| *number* | Integer, expressed in equipment impairment factor units, specifying the ICPIF value. Valid entries are 0 to 55. |

### Default

30 equipment impairment factor units.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **icpif** command to specify the maximum acceptable impairment factor for the voice calls sent by the selected dial peer.

This command only applies to VoIP peers.

### Example

The following example disables the **icpif** command:

```
dial-peer voice 10 voip
 icpif 0
```

# impedance

To specify the terminating impedance of a voice-port interface, use the **impedance** voice-port configuration command. Use the **no** form of this command to restore the default value.

**impedance** {**600c** | **600r** | **900c** | **complex1** | **complex2**}
**no impedance**

### Syntax Description

| | |
|---|---|
| **600c** | 600 ohms complex. |
| **600r** | 600 ohms real. |
| **900c** | 900 ohms complex. |
| complex1 | Complex 1. |
| complex2 | Complex 2. |

### Default

600 ohms.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **impedance** command to specify the terminating impedance of an FXO voice-port interface. The impedance value selected needs to match the specifications from the specific telephony system to which it is connected. Different countries often have different standards for impedance. CO switches in the United States are predominantly 600r. PBXs in the United States are normally either 600r or 900c.

If the impedance is set incorrectly (if there is an impedance mismatch), a significant amount of echo is generated (which could be masked if the **echo-cancel** command has been enabled). In addition, gains might not work correctly if there is an impedance mismatch.

Configuring the impedance on a voice port changes the impedance on both voice ports of a VIC. This voice port must be shut down and then opened for the new value to take effect.

This command applies to FXS, FXO, and E&M voice ports.

### Example

The following example configures an FXO voice port for a terminating impedance of 600 ohms:

```
voice port 0/0
impedance 600r
```

# input gain

To configure a specific input gain value, use the **input gain** voice-port configuration command. Use the **no** form of this command to disable this feature.

**input gain** *value*
**no input gain** *value*

### Syntax Description

| | |
|---|---|
| *value* | Amount of gain in decibels (dB) to be inserted at the receiver side of the interface. Acceptable value is any integer from –6 to 14. |

### Default

0 dB.

### Command Mode

Voice-port configuration.

### Usage Guidelines

A system-wide loss plan must be implemented using both **input gain** and **output attenuation** commands. Other equipment (including PBXs) in the system must be taken into account when creating a loss plan. The default value for this command assumes that a standard transmission loss plan is in effect, meaning that, normally, there must be –6 dB of attenuation between phones. Connections are implemented to provide –6 dB of attenuation when the **input gain** and **output attenuation** commands are configured with the default value of 0.

You cannot increase the gain of a signal going out into the PSTN, but you can decrease it. Therefore, if the voice level is too high, you can decrease the volume by either decreasing the input gain value or by increasing the output attenuation.

You can increase the gain of a signal coming into the router. If the voice level is too low, you can increase the input gain.

### Example

The following example configures a 3-dB gain for the receiver side of the interface:

```
voice port 0/0
input gain 3
```

### Related Commands

**output attenuation**

# ip precedence

To set IP precedence (priority) for packets sent by the dial peer, use the **ip precedence** dial-peer configuration command. Use the **no** form of this command to restore the default value for this command.

**ip precedence** *number*
**no ip precedence**

### Syntax Description

| | |
|---|---|
| *number* | Integer specifying the IP precedence value. Valid entries are 0 to 7. A value of 0 means that no precedence (priority) has been set. |

### Default

No precedence (0).

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **ip precedence** command to configure the value set in the IP precedence field when voice data packets are sent over the IP network. This command should be used if the IP link utilization is high and the QoS for voice packets need to have a higher priority than other IP packets. The **ip precedence** command should also be used if RSVP is not enabled and the user would like to give voice packets a higher priority over other IP data traffic.

This command only applies to VoIP peers.

### Example

The following example sets the IP precedence at 5:

```
dial-peer voice 10 voip
 ip precedence 5
```

# ip udp checksum

To calculate the User Datagram Protocol (UDP) checksum for voice packets transmitted by the dial peer, use the **ip udp checksum** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**ip udp checksum**
**no ip udp checksum**

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **ip udp checksum** command to enable UDP checksum calculation for each outbound voice packet. This command is disabled by default to speed up the transmission of the voice packets. If you suspect that the connection has a high error rate, you should enable **ip udp checksum** to prevent bad voice packets forwarded to the DSP.

This command only applies to VoIP peers.

### Example

The following example calculates the UDP checksum for voice packets transmitted by this dial peer:

```
dial-peer voice 10 voip
 ip udp checksum
```

# music-threshold

To specify the threshold for on-hold music for a specified voice port, use the **music-threshold** voice-port configuration command. Use the **no** form of this command to disable this feature.

**music-threshold** *number*
**no music-threshold** *number*

### Syntax Description

| | |
|---|---|
| *number* | On-hold music threshold in dB. Valid entries are any integer from –70 to –30. |

### Default

–38 dB.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **music-threshold** command to specify the dB level of music played when calls are on hold. This command tells the firmware to pass steady data above the specified level. It only affects the operation of VAD when receiving voice.

If the value for this command is set too high, VAD interprets music-on-hold as silence, and the remote end does not hear the music. If the value for this command is set too low, VAD compresses and passes silence when the background is noisy, creating unnecessary voice traffic.

### Example

The following sets the dB threshold for the music played when calls are put on hold to –35:

```
voice port 0/0
 music-threshold –35
```

# non-linear

To enable nonlinear processing in the echo canceller, use the **non-linear** voice-port configuration command. Use the **no** form of this command to disable this feature.

**non-linear**
**no non-linear**

### Syntax Description

This command has no arguments or keywords.

### Default

Enabled.

### Command Mode

Voice-port configuration.

### Usage Guidelines

This command is associated with the echo canceller operation. The **echo-cancel enable** command must be enabled for the **non-linear** command to take effect. Use the **non-linear** command to shut off any signal if no near-end speech is detected.

Enabling the **non-linear** command normally improves performance, although some users might hear truncation of consonants at the end of sentences when this command is enabled.

This feature is also generally known as residual echo suppression.

### Example

The following example enables nonlinear call processing:

```
voice port 0/0
 non-linear
```

### Related Commands

**echo-cancel enable**

# num-exp

To define how to expand an extension number into a particular destination pattern, use the **num-exp** global configuration command.

**num-exp** *extension-number expanded-number*

### Syntax Description

| | |
|---|---|
| *extension-number* | Digit(s) defining an extension number for a particular dial peer. |
| *expanded-number* | Digit(s) defining the expanded telephone number or destination pattern for the extension number listed. |

### Default

No number expansions are predefined.

### Command Mode

Global configuration.

### Usage Guidelines

Use the **num-exp** global configuration command to define how to expand a particular set of numbers (for example, an extension number) into a particular destination pattern. With this command, you can map specific extensions and expanded numbers together by explicitly defining each number, or you can define extensions and expanded numbers by using variables. You can also use this command to convert seven-digit numbers to numbers of less than seven digits.

Use a period (.) as a variable or wildcard representing a single number. Use a separate period for each number you want to represent with a wildcard—meaning that if you want to replace four numbers in an extension with wildcards, enter four periods.

### Examples

The following example expands the extension number *54001* to *14085554001*:

```
num-exp 54001 14085554001
```

The following example shows how to expand all five-digit extensions beginning with 5 and append the extension numbers to *1408555*:

```
num-exp 5.... 1408555....
```

# operation

To select a specific cabling scheme for E&M ports, use the **operation** voice-port configuration command. Use the **no** form of this command as an alternative method of configuring two-wire operation.

**operation** {**2-wire** | **4-wire**}
**no operation** {**2-wire** | **4-wire**}

### Syntax Description

| | |
|---|---|
| **2-wire** | Two-wire E&M cabling scheme. |
| 4-wire | Four-wire E&M cabling scheme. |

### Default

**2-wire**.

### Command Mode

Voice-port configuration.

### Usage Guidelines

The **operation** command only affects voice traffic. Signaling is independent of two-wire versus four-wire settings. If the wrong cable scheme is specified, the user might get voice traffic in only one direction.

Configuring the **operation** command on a voice port changes the operation of both voice ports on a VIC. The voice port must be shut down and then opened again for the new value to take effect.

This command does not apply to FXS or FXO interfaces because those are, by definition, two-wire interfaces.

### Example

The following example specifies that an E&M port uses a four-wire cabling scheme:

```
voice port 0/0
 operation 4-wire
```

# output attenuation

To configure a specific output attenuation value, use the **output attenuation** voice-port configuration command. Use the **no** form of this command to disable this feature.

**output attenuation** *value*
**no output attenuation**

### Syntax Description

| | |
|---|---|
| *value* | Amount of attenuation in dB at the transmit side of the interface. Acceptable value is any integer from 0 to 14. |

### Default

0 dB.

### Command Mode

Voice-port configuration.

### Usage Guidelines

A system-wide loss plan must be implemented by using both **input gain** and **output attenuation** commands. Other equipment (including PBXs) in the system must be taken into account when creating a loss plan. The default value for this command assumes that a standard transmission loss plan is in effect, meaning that, normally, there must be –6 dB of attenuation between phones. Connections are implemented to provide –6 dB of attenuation when the **input gain** and **output attenuation** commands are configured with the default value of 0.

You cannot increase the gain of a signal going out into the PSTN, but you can decrease it. Therefore, if the voice level is too high, you can decrease the volume by either decreasing the input gain value or by increasing the output attenuation.

### Example

The following example configures a 3-dB gain to be inserted at the transmit side of the interface:

```
voice port 0/0
 output attenuation 3
```

### Related Commands

**input gain**

# port

To associate a dial peer with a specific voice port, use the **port** dial-peer configuration command. Use the **no** form of this command to cancel this association.

**port** *slot-number*/*port*
**no port**

### Syntax Description

| | |
|---|---|
| *slot-number* | Slot number in the router where the VIC is installed. Valid entries are from 0 to 2, depending on the slot where it has been installed. |
| *port* | Voice port. Valid entries are 0 or 1. |

### Default

No port is preconfigured.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **port** configuration command to associate the designated voice port with the selected dial peer.

This command is used for calls incoming from a telephony interface to select an incoming dial peer and for calls coming from the VoIP network to match a port with the selected outgoing dial peer.

This command only applies to POTS peers.

### Example

The following example associates a dial peer with slot 0 and access through port 0:

```
dial-peer voice 10 pots
 port 0/0
```

# prefix

To specify the prefix of the dialed digits for this dial peer, use the **prefix** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**prefix** *string*
**no prefix**

### Syntax Description

| | |
|---|---|
| *string* | Integers representing the prefix of the telephone number associated with the specified dial peer. Valid numbers are 0 through 9, and a comma (**,**). Use a comma to include a pause in the prefix. |

### Default

Null string.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **prefix** command to specify a prefix for a specific dial peer. When an outgoing call is initiated to this dial peer, the **prefix** *string* value is first sent to the telephony interface, before the telephone number is associated with the dial peer.

If you want to configure different prefixes for dialed numbers on the same interface, you need to configure different dial peers.

This command only applies to POTS peers.

### Example

The following example specifies a prefix of *9* and then a pause:

```
dial-peer voice 10 pots
 prefix 9,
```

### Related Commands

**answer-address**
**destination-pattern**

# req-qos

To specify the desired QoS to be used in reaching a specified dial peer, use the **req-qos** dial-peer configuration command. Use the **no** form of this command to restore the default value for this command.

**req-qos** {**best-effort** | **controlled-load** | **guaranteed-delay**}
**no req-qos**

### Syntax Description

| | |
|---|---|
| **best-effort** | RSVP makes no bandwidth reservation. |
| **controlled-load** | RSVP guarantees a single level of preferential service, presumed to correlate to a delay boundary. The controlled load service uses admission (or capacity) control to ensure that preferential service is received even when the bandwidth is overloaded. |
| **guaranteed-delay** | RSVP reserves bandwidth and guarantees a minimum bit rate and preferential queuing if the bandwidth reserved is not exceeded. |

### Default

**best-effort**. The **no** form of this command restores the default value.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **req-qos** command to request a specific QoS to be used in reaching a dial peer. This command is like **acc-qos**; the software reserves a certain amount of bandwidth to provide the selected QoS. Cisco IOS software uses RSVP to request QoS guarantees from the network.

This command only applies to VoIP peers.

### Example

The following example configures guaranteed-delay as the desired (requested) QoS to a dial peer:

```
dial-peer voice 10 voip
 req-qos guaranteed-delay
```

### Related Commands

**acc-qos**

# ring frequency

To specify the ring frequency for a specified FXS voice port, use the **ring frequency** voice-port configuration command. Use the **no** form of this command to reset the default value for this command.

**ring frequency** *number*
**no ring frequency**

### Syntax Description

| | |
|---|---|
| *number* | Ring frequency in Hz used in the FXS interface. Valid entries are 25 and 50 Hz. |

### Default

25 Hz.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **ring frequency** command to select a specific ring frequency for an FXS voice port. Use the **no** form of this command to reset the default value. The ring frequency you select must match the connected equipment. If set incorrectly, the attached phone might not ring or might buzz. In addition, the ring frequency is usually country-dependent, and you should take into account the appropriate ring frequency for your area before configuring this command.

This command does not affect ringback, which is the ringing a user hears when placing a remote call.

### Example

The following example configures the ring frequency for 50 Hz:

```
voice port 0/0
 ring frequency 50
```

### Related Commands

**ring number**

# ring number

To specify the number of rings for a specified FXO voice port, use the **ring number** voice-port configuration command. Use the **no** form of this command to reset the default value for this command.

**ring number** *number*
**no ring number** *number*

### Syntax Description

| | |
|---|---|
| *number* | Number of rings detected before answering the call. Valid entries are numbers from 1 to 10. |

### Default

1 ring.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **ring number** command to set the maximum number of rings to be detected before answering a call over an FXO voice port. Use the **no** form of this command to reset the default value.

Normally, this command should be set to the default so that incoming calls are answered quickly. If you have other equipment available on the line to answer incoming calls, you might want to set the value higher to give the equipment sufficient time to respond. In that case, the FXO interface would answer if the other equipment on line did not answer the incoming call in the configured number of rings.

This command does not apply to FXS or E&M interfaces because they do not receive ringing to receive a call.

### Example

The following example sets five rings as the maximum number of rings to be detected before closing a connection over this voice port:

```
voice port 0/0
 ring number 5
```

### Related Commands

**ring frequency**

# session protocol

To establish a session protocol for calls between the local and remote routers via the packet network, use the **session protocol** dial-peer configuration command. Use the **no** form of this command to reset the default value for this command.

**session protocol cisco**
**no session protocol**

### Syntax Description

| | |
|---|---|
| **cisco** | Cisco Session Protocol. |

### Default

**cisco**.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

For this release, **cisco** is the only applicable session protocol. This command only applies to VoIP peers.

### Example

The following example selects Cisco Session Protocol as the session protocol:

```
dial-peer voice 10 voip
 session protocol cisco
```

### Related Commands

**session target**

# session target

To specify a network-specific address for a specified dial peer, use the **session target** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**session target** {**ipv4:***destination-address* | **dns:**[**$s$. | $d$. | $u$.**] *host-name* | **loopback:rtp** | **loopback:compressed** | **loopback:uncompressed**}
**no session target**

### Syntax Description

| | |
|---|---|
| **ipv4:***destination-address* | IP address of the dial peer. |
| **dns:**host-name | Domain name system (DNS) server is used to resolve the name of the IP address. Valid entries for this parameter are characters representing the name of the host device.<br>(Optional) You can use one of the following wildcards with this keyword when defining the session target for VoIP dial peers:<br>• **$s$.**—Source destination pattern is used as part of the domain name.<br>• **$d$.**—Destination number is used as part of the domain name.<br>• **$u$.**—Unmatched portion of the destination pattern (such as a defined extension number) is used as part of the domain name. |
| **loopback:rtp** | All voice data is looped-back to the originating source. This only applies to VoIP dial peers. |
| **loopback:compressed** | All voice data is looped-back in compressed mode to the originating source. This only applies to POTS dial peers. |
| **loopback:uncompressed** | All voice data is looped-back in uncompressed mode to the originating source. This only applies to POTS dial peers. |

### Default

Enabled with no IP address or domain name defined.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **session target** command to specify a network-specific address or domain name for a dial peer.

The **session target loopback** command is used for testing the voice transmission path of a call. The loopback point depends on the call origination and the loopback type selected.

The **session target dns** command can be used with or without the specified wildcards. The optional wildcards reduce the number of VoIP dial-peer session targets you need to configure if you have groups of numbers associated with a particular router.

### Example

The following example configures a session target using **dns** for hostname *voice_router* in the domain *cisco.com*:

```
dial-peer voice 10 voip
 session target dns:voice_router.cisco.com
```

The following example configures a session target using **dns** and the optional **$u$.** wildcard. In this example, the destination pattern has been configured to allow for any four-digit extension, beginning with the numbers *1310222*. The optional wildcard **$u$.** means that the router uses the unmatched portion of the dialed number—in this case, the four-digit extension—to identify the dial peer. As in the previous example, the domain is *cisco.com*.

```
dial-peer voice 10 voip
 destination-pattern 1310222....
 session target dns:$u$.cisco.com
```

The following example configures a session target using **dns**, with the optional **$d$.** wildcard. In this example, the destination pattern has been configured for *13102221111*. The optional wildcard **$d$.** means that the router uses the destination pattern to identify the dial peer in the *cisco.com* domain.

```
dial-peer voice 10 voip
 destination-pattern 13102221111
 session target dns:$d$.cisco.com
```

### Related Commands

**destination-pattern**
**session protocol**

# show call active voice

To show the active call table, use the **show call active voice** privileged EXEC command.

**show call active voice**

### Syntax Description

This command contains no arguments or keywords.

### Command Mode

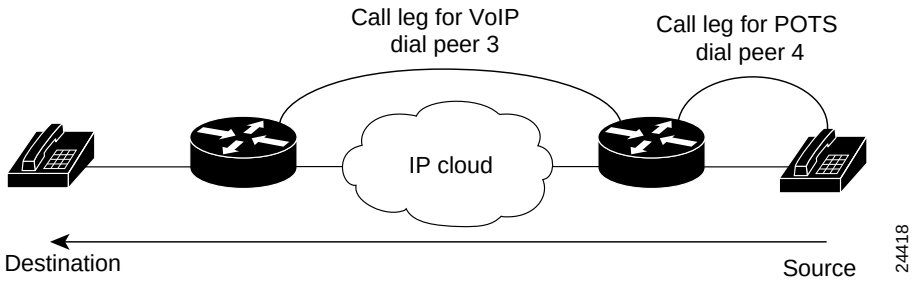
Privileged EXEC.

### Usage Guidelines

Use the **show call active voice** privileged EXEC command to display the contents of the active call table, which shows all of the calls currently connected through the router.

For each call, there are two call legs, a POTS call leg and a VoIP call leg. A call leg is a discrete segment of a call between two points in the connection. Each dial peer creates a call leg, as shown in Figure 1.

*Figure 1 Call Legs Example*




These two call legs are associated by the connection ID. The connection ID is global across the voice network so that you can associate two call legs on one router with two call legs on another router, thereby providing an end-to-end view of a call.

## Sample Display

The following is sample output from the **show call active voice** command:

```
router# show call active voice
GENERIC: SetupTime=21072 Index=0 PeerAddress= PeerSubAddress= PeerId=0
PeerIfIndex=0 LogicalIfIndex=0 ConnectTime=0 CallState=3 CallOrigin=2 ChargedUnits=0
InfoType=0 TransmitPackets=375413 TransmitBytes=7508260 ReceivePackets=377734
ReceiveBytes=7554680

VOIP: ConnectionId[0x19BDF910 0xAF500007 0x0 0x58ED0] RemoteIPAddress=17635075
RemoteUDPPort=16394 RoundTripDelay=0 SelectedQoS=0 SessionProtocol=1
SessionTarget= OnTimeRvPlayout=0 GapFillWithSilence=0 GapFillWithPrediction=600
GapFillWithInterpolation=0 GapFillWithRedundancy=0 HiWaterPlayoutDelay=110
LoWaterPlayoutDelay=64 ReceiveDelay=94 VADEnable=0 CoderTypeRate=0

GENERIC: SetupTime=21072 Index=1 PeerAddress=14085554001 PeerSubAddress=
PeerId=0 PeerIfIndex=0 LogicalIfIndex=5 ConnectTime=21115 CallState=4 CallOrigin=1
ChargedUnits=0 InfoType=1 TransmitPackets=377915 TransmitBytes=7558300
ReceivePackets=375594 ReceiveBytes=7511880

TELE: ConnectionId=[0x19BDF910 0xAF500007 0x0 0x58ED0] TxDuration=16640
VoiceTxDuration=16640 FaxTxDuration=0 CoderTypeRate=0 NoiseLevel=0 ACOMLevel=4
OutSignalLevel=-440 InSignalLevel=-440 InfoActivity=2 ERLLevel=227
SessionTarget=
```

Table 3 provides an alphabetical listing of the fields in this output and a description of each field.

*Table 3 Show-Call-Active-Voice Command Field Descriptions*

| Field | Description |
|---|---|
| ACOM Level | Current ACOM level for the call. This value is sum of the Echo Return Loss, Echo Return Loss Enhancement, and nonlinear processing loss for the call. |
| CallOrigin | Call origin; answer versus originate. |
| CallState | Current state of the call. |
| CoderTypeRate | Negotiated coder transmit rate of voice/fax compression during the call. |
| ConnectionId | Global call identifier of a gateway call. |

*Table 3 Show-Call-Active-Voice Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| ConnectTime | Time at which the call was connected. |
| Dial-Peer | Tag of the dial peer transmitting this call. |
| ERLLevel | Current Echo Return Loss (ERL) level for this call. |
| FaxTxDuration | Duration of fax transmission from this peer to voice gateway for this call. You can derive the Fax Utilization Rate by dividing the FaxTxDuration value by the TxDuration value. |
| GapFillWith Silence | Duration of voice signal replaced with silence because voice data was lost or not received on time for this call. |
| GapFillWithPrediction | Duration of voice signal played out with signal synthesized from parameters or samples of data preceding in time because voice data was lost or not received in time from the voice gateway for this call. An example of such pullout is frame-eraser or frame-concealment strategies in G.729 and G.723.1 compression algorithms. |
| GapFillWithInterpolation | Duration of voice signal played out with signal synthesized from parameters or samples of data preceding and following in time because voice data was lost or not received on time from voice gateway for this call. |
| GapFillWith Redundancy | Duration of voice signal played out with signal synthesized from redundancy parameters available because voice data was lost or not received on time from voice gateway for this call. |
| HiWaterPlayoutDelay | High-water mark Voice Playout FIFO Delay during this call. |
| Index | Dial-peer identification number. |
| InfoActivity | Active information transfer activity state for this call. |
| InfoType | Information type for this call. |
| InSignalLevel | Active input signal level from the telephony interface used by this call. |
| LogicalIfIndex | Index number of the logical interface for this call. |
| LoWaterPlayoutDelay | Low-water mark Voice Playout FIFO Delay during the call. |
| NoiseLevel | Active noise level for the call. |
| OnTimeRvPlayout | Duration of voice playout from data received on time for this call. You can derive the Total Voice Playout Duration for Active Voice by adding the OnTimeRvPlayout value to the GapFill values. |
| OutSignalLevel | Active output signal level to telephony interface used by this call. |
| PeerAddress | Destination pattern associated with this peer. |
| PeerId | ID value of the peer table entry to which this call was made. |
| PeerIfIndex | Voice-port index number for this peer. |
| PeerSubaddress | Subaddress to which this call is connected. |
| ReceiveBytes | Number of bytes received by the peer during this call. |
| ReceiveDelay | Average Playout FIFO Delay plus the decoder delay during the voice call. |
| ReceivePackets | Number of packets received by this peer during this call. |

*Table 3 Show-Call-Active-Voice Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| RemoteIPAddress | Remote system IP address for the VoIP call. |
| RemoteUDPPort | Remote system UDP listener port to which voice packets are transmitted. |
| RoundTripDelay | Voice packet round trip delay between the local and remote system on the IP backbone during the call. |
| SelectedQoS | Selected RSVP QoS for the call. |
| SessionProtocol | Session protocol used for an Internet call between the local and remote router via the IP backbone. |
| SessionTarget | Session target of the peer used for the call. |
| SetupTime | Value of the System UpTime when the call associated with this entry was started. |
| TransmitBytes | Number of bytes transmitted from this peer during the call. |
| TransmitPackets | Number of packets transmitted from this peer during the call. |
| TxDuration | Duration of transmit path open from this peer to the voice gateway for the call. |
| VADEnable | Whether or not VAD was enabled for this call. |
| VoiceTxDuration | Duration of voice transmission from this peer to voice gateway for this call. You can derive the Voice Utilization Rate by dividing the VoiceTxDuration value by the TxDuration value. |

### Related Commands

**show call history voice**
**show dial-peer voice**
**show num-exp**
**show voice port**

# show call history voice

To display the call history table, use the **show call history voice** privileged EXEC command.

**show call history voice last** *number*

### Syntax Description

| | |
|---|---|
| last *number* | Displays the last calls connected, where the number of calls displayed is defined by the argument *number.* Valid entries for the argument *number* is any number from 1 to 2147483647. |

### Command Mode

Privileged EXEC.

## Usage Guidelines

Use the **show call history voice** privileged EXEC command to display the call history table. The call history table contains a listing of all calls connected through this router in descending time order since VoIP was enabled. You can display subsets of the call history table by using specific keywords. To display the last calls connected through this router, use the keyword **last**, and define the number of calls to be displayed with the argument *number*.

## Sample Display

The following is sample output from the **show call history voice** command:

```
router# show call history voice
GENERIC: SetupTime=20405 Index=0 PeerAddress= PeerSubAddress= PeerId=0
PeerIfIndex=0 LogicalIfIndex=0 DisconnectCause=NORMAL DisconnectText= ConnectTime=0
DisconectTime=20595 CallOrigin=2 ChargedUnits=0 InfoType=0 TransmitPackets=0
TransmitBytes=0 ReceivePackets=0 ReceiveBytes=0

VOIP: ConnectionId[0x19BDF910 0xAF500006 0x0 0x56590] RemoteIPAddress=17635075
RemoteUDPPort=16392 RoundTripDelay=0 SelectedQoS=0 SessionProtocol=1
SessionTarget= OnTimeRvPlayout=0 GapFillWithSilence=0 GapFillWithPrediction=0
GapFillWithInterpolation=0 GapFillWithRedundancy=0 HiWaterPlayoutDelay=0
LoWaterPlayoutDelay=0 ReceiveDelay=0 VADEnable=0 CoderTypeRate=0

TELE: ConnectionId=[0x19BDF910 0xAF500006 0x0 0x56590] TxDuration=3030
VoiceTxDuration=2700 FaxTxDuration=0 CoderTypeRate=0 NoiseLevel=0 ACOMLevel=0
SessionTarget=
```

Table 4 provides an alphabetical listing of the fields in this output and a description of each field.

***Table 4 Show-Call-History-Voice Command Field Descriptions***

| Field | Description |
|---|---|
| ACOMLevel | Average ACOM level for this call. This value is sum of the Echo Return Loss, Echo Return Loss Enhancement, and nonlinear processing loss for the call. |
| CallOrigin | Call origin; answer versus originate. |
| CoderTypeRate | Negotiated coder rate. This value specifies the transmit rate of voice/fax compression to its associated call leg for the call. |
| ConnectionID | Global call identifier for the gateway call. |
| ConnectTime | Time the call was connected. |
| DisconnectCause | Description explaining why the call was disconnected. |
| DisconnectText | Descriptive text explaining the disconnect reason. |
| DisconnectTime | Time the call was disconnected. |
| FaxDuration | Duration of fax transmitted from this peer to the voice gateway for this call. You can derive the Fax Utilization Rate by dividing this value by the TxDuration value. |
| GapFillWithSilence | Duration of voice signal replaced with silence because the voice data was lost or not received on time for this call. |

*Table 4 Show-Call-History-Voice Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| GapFillWithPrediction | Duration of voice signal played out with signal synthesized from parameters or samples of data preceding and following in time because the voice data was lost or not received on time from the voice gateway for this call. |
| GapFillWithInterpolation | Duration of voice signal played out with signal synthesized from parameters or samples of data preceding and following in time because the voice data was lost or not received on time from the voice gateway for this call. |
| GapFillWithRedundancy | Duration of voice signal played out with signal synthesized from redundancy parameters available because the voice data was lost or not received on time from the voice gateway for this call. |
| HiWaterPlayoutDelay | High-water mark Voice Playout FIFO Delay during the voice call. |
| Index | Index number identifying the voice-peer for this call. |
| InfoType | Information type for this call. |
| LogicalIfIndex | Index of the logical voice port for this call. |
| LoWaterPlayoutDelay | Low-water mark Voice Playout FIFO Delay during the voice call. |
| NoiseLevel | Average noise level for this call. |
| OnTimeRvPlayout | Duration of voice playout from data received on time for this call. You can derive the Total Voice Playout Duration for Active Voice by adding the OnTimeRvPlayout value to the GapFill values. |
| PeerAddress | Destination pattern or number to which this call is connected. |
| PeerId | ID value of the peer entry table to which this call was made. |
| PeerIfIndex | Index number of the logical interface through which this call was made. For ISDN media, this would be the index number of the B channel used for the call. |
| PeerSubAddress | Subaddress to which this call is connected. |
| ReceiveBytes | Number of bytes received by the peer during this call. |
| ReceiveDelay | Average Playout FIFO Delay plus the decoder delay during the voice call. |
| ReceivePackets | Number of packets received by this peer during the call. |
| RemoteIPAddress | Remote system IP address for the call. |
| RemoteUDPPort | Remote system UDP listener port to which voice packets for this call are transmitted. |
| RoundTripDelay | Voice packet round trip delay between the local and remote system on the IP backbone for this call. |
| SelectedQoS | Selected RSVP QoS for the call. |
| Session Protocol | Session protocol to be used for an Internet call between the local and remote router via the IP backbone. |
| Session Target | Session target of the peer used for the call. |
| SetUpTime | Value of the System UpTime when the call associated with this entry was started. |

*Table 4 Show-Call-History-Voice Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| TransmitBytes | Number of bytes transmitted by this peer during the call. |
| TransmitPackets | Number of packets transmitted by this peer during the call. |
| TxDuration | Duration of the transmit path open from this peer to the voice gateway for the call. |
| VADEnable | Whether or not VAD was enabled for this call. |
| VoiceTxDuration | Duration of voice transmitted from this peer to voice gateway for this call. You can derive the Voice Utilization Rate by dividing the VoiceTxDuration by the TxDuration value. |

### Related Commands

**show call active voice**
**show dial-peer voice**
**show num-exp**
**show voice port**

# show controllers voice

To display information about voice related hardware, use the **show controllers voice** privileged EXEC command.

**show controllers voice**

### Syntax Description

This command contains no arguments or keywords.

### Command Mode

Privileged EXEC.

### Usage Guidelines

This command displays interface status information that is specific to voice related hardware, such as, the registers of the TDM switch, the host port interface of the DSP, and the DSP firmware versions.

The information displayed is generally useful for diagnostic tasks performed by technical support people only.

### Sample Display

The following is sample output from the **show controllers voice** command:

```
router# show controllers voice
EPIC Switch registers:
STDA 0xFF STDB 0x0 SARA 0x0 SARB 0xFF SAXA 0xFF SAXB 0x0 STCR 0x3F MFAIR 0x3F
STAR 0x65 OMDR 0xE2 VNSR 0x0 PMOD 0x4C PBNR 0xFF POFD 0xF0 POFU 0x18
PCSR 0x1 PICM 0x0 CMD1 0xA0 CMD2 0x70 CBNR 0xFF CTAR 0x2 CBSR 0x20 CSCR 0x0

DSP 0 Host Port Interface:
HPI Control Register 0x202
InterfaceStatus 0x2A MaxMessageSize 0x80
RxRingBufferSize 0x6 TxRingBufferSize 0x9
pInsertRx 0x1 pRemoveRx 0x1 pInsertTx 0x2 pRemoveTx 0x2

Rx Message 0:
packet_length 12 channel_id 0 packet_id 6 process id1 0xFECE process id2 0xFACE
0000:0000

Rx Message 1:
packet_length 12 channel_id 0 packet_id 6 process id1 0xFECE process id2 0xFACE
0000:0000

Rx Message 2:
packet_length 12 channel_id 0 packet_id 6 process id1 0xFECE process id2 0xFACE
0000:0000

 --More--         Rx Message 3:
packet_length 12 channel_id 0 packet_id 6 process id1 0xFECE process id2 0xFACE
0000:0000

Rx Message 4:
packet_length 12 channel_id 0 packet_id 6 process id1 0xFECE process id2 0xFACE
0000:0000

Rx Message 5:
packet_length 12 channel_id 0 packet_id 6 process id1 0xFECE process id2 0xFACE
0000:0000

Tx Message 0:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 0042 003F 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Tx Message 1:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 0043 0040 0000 0000 0000 0000
 --More--         0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Tx Message 2:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 003B 0038 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Tx Message 3:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 003C 0039 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000
```

```
Tx Message 4:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 003D 003A 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000
 --More--
Tx Message 5:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 003E 003B 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Tx Message 6:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 003F 003C 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Tx Message 7:
packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process id2 0xFACE
0000:0000 0000 0000 0000 0040 003D 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Tx Message 8:
 --More--         packet_length 66 channel_id 0 packet_id 198 process id1 0xFECE process
id2 0xFACE
0000:0000 0000 0000 0000 0041 003E 0000 0000 0000 0000
0020:0000 0006 0006 0006 0006 0006 0006 0006 0006 0006
0040:0006 0006 0006 0006 0006 0006 0006 0000

Bootloader 1.8, Appn 3.1
Application firmware 3.1.1, Built by claux on Mon Mar 22 16:32:13 1999

VIC Interface Foreign Exchange Station 1/0, DSP instance (0x19355C0)
 Singalling channel num 128 Signalling proxy 0x0 Signaling dsp 0x19355C0
 tx outstanding 0, max tx outstanding 32
 ptr 0x0, length 0x0, max length 0x0
 dsp_number 0, Channel ID 1
 received 0 packets, 0 bytes, 0 gaint packets
 0 drops, 0 no buffers, 0 input errors 0 input overruns
 264434 bytes output, 1036 frames output, 0 output errors, 0 output underrun
 0 unaligned frames

VIC Interface Foreign Exchange Station 1/1, DSP instance (0x19357F0)
 Singalling channel num 129 Signalling proxy 0x0 Signaling dsp 0x19357F0
 tx outstanding 0, max tx outstanding 32
 ptr 0x0, length 0x0, max length 0x0
 --More--         dsp_number 0, Channel ID 2
 received 0 packets, 0 bytes, 0 gaint packets
 0 drops, 0 no buffers, 0 input errors 0 input overruns
 68 bytes output, 4 frames output, 0 output errors, 0 output underrun
 0 unaligned frames
```

# show diag

To display hardware information for the router, use the **show diag** privileged EXEC command.

**show diag**

### Syntax Description

This command contains no arguments or keywords.

### Command Mode

Privileged EXEC.

### Usage Guidelines

This command displays information for the electrically erasable programmable read-only memory (EEPROM), motherboard, and the WAN interface cards and voice interface cards (WICs/VICs).

## Sample Display

The following is sample output from the **show diag** command:

```
router# show diag
Slot 0:
        C1750 1FE VE Mainboard port adapter, 6 ports
        Port adapter is analyzed
        Port adapter insertion time unknown
        EEPROM contents at hardware discovery:
        Hardware revision 0.0           Board revision UNKNOWN
        Serial number     1314672220    Part number    00-0000-00
        Test history      0x0           RMA number     00-00-00
        EEPROM format version 1
        EEPROM contents (hex):
          0x20:01 C9 00 00 4E 5C 4E 5C 00 00 00 00 00 00 00 00
          0x30:00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00

Packet Voice DSP Module:
        Hardware Revision        :1.0
        Board Revision           :01
        Processor type           :02
        Part Number              :73-3933-01
        Number of DSP's          :2
        Type of DSP              :TMS320C549
        EEPROM format version 4
        EEPROM contents (hex):
        0x00:  04 FF 40 01 5B 41 01 00 42 30 31 09 02 82 49 0F
        0x10:  5D 01 FF

        WIC Slot 0:
        BRI U - 2091 WAN daughter card
        Hardware revision 1.3   Board revision A0
        Serial number     0004147773    Part number    800-01834-01
        Test history      0x00          RMA number     00-00-00
        Connector type    WAN Module
        EEPROM format version 1
        EEPROM contents (hex):
        0x20:  01 09 01 03 00 3F 4A 3D 50 07 2A 01 00 00 00 00
        0x30:  50 00 00 00 96 11 06 01 FF FF FF FF FF FF FF FF

        WIC Slot 1:
        Dual FXS Voice Interface Card WAN daughter card
        Hardware revision 1.1   Board revision C0
        Serial number     0010377882    Part number    800-02493-01
        Test history      0x00          RMA number     00-00-00
        Connector type    WAN Module
        EEPROM format version 1
        EEPROM contents (hex):
        0x20:  01 0E 01 01 00 9E 5A 9A 50 09 BD 01 00 00 00 00
        0x30:  60 00 00 00 98 09 10 01 FF FF FF FF FF FF FF FF

        WIC Slot 2:
        Dual EAM Voice Interface Card WAN daughter card
        Hardware revision 1.1   Board revision C0
        Serial number     0009886880    Part number    800-02497-01
        Test history      0x00          RMA number     00-00-00
        Connector type    WAN Module
        EEPROM format version 1
        EEPROM contents (hex):
        0x20:  01 0F 01 01 00 96 DC A0 50 09 C1 01 00 00 00 00
        0x30:  60 00 00 00 98 08 26 01 FF FF FF FF FF FF FF FF
Message-ID:<37014A10.3506648@cisco.com>
```

# show dial-peer voice

To display configuration information for dial peers, use the **show dial-peer voice** privileged EXEC command.

**show dial-peer voice** [*number*]

## Syntax Description

| | |
|---|---|
| *number* | Displays configuration for the dial peer identified by the argument *number*. Valid entries are any integers that identify a specific dial peer, from 1 to 32767. |

## Command Mode

Privileged EXEC.

## Usage Guidelines

Use the **show dial-peer voice** privileged EXEC command to display the configuration for all VoIP and POTS dial peers configured for the router. To show configuration information for only one specific dial peer, use the argument *number* to identify the dial peer.

## Sample Display

The following is sample output from the **show dial-peer voice** command for a POTS dial peer:

```
router# show dial-peer voice 1
VoiceEncapPeer1
        tag = 1, dest-pat = `14085551000',
        answer-address = `',
        group = 0, Admin state is up, Operation state is down
        Permission is Both,
        type = pots, prefix = `',
        session target = `', voice port =
        Connect Time = 0, Charged Units = 0
        Successful Calls = 0, Failed Calls = 0
        Accepted Calls = 0, Refused Calls = 0
        Last Disconnect Cause is “”
        Last Disconnect Text is “”
        Last Setup Time = 0
```

The following is sample output from the **show dial-peer voice** command for a VoIP dial peer:

```
router# show dial-peer voice 10
VoiceOverIpPeer10
        tag = 10, dest-pat = `',
        incall-number = `14085',
        group = 0, Admin state is up, Operation state is down
        Permission is Answer,
        type = voip, session target = `',
        sess-proto = cisco, req-qos = bestEffort,
        acc-qos = bestEffort,
        fax-rate = voice, codec = g729r8,
        Expect factor = 10,Icpif = 30, VAD = disabled, Poor QOV Trap = disabled,
        Connect Time = 0, Charged Units = 0
        Successful Calls = 0, Failed Calls = 0
        Accepted Calls = 0, Refused Calls = 0
        Last Disconnect Cause is “”
        Last Disconnect Text is “”
        Last Setup Time = 0
```

Table 5 explains the fields contained in both of these examples.

***Table 5 Show-Dial-Peer-Voice Command Field Descriptions***

| Field | Description |
|---|---|
| AcceptedCalls | Number of calls from this peer accepted since system startup. |
| acc-qos | Lowest acceptable QoS configured for calls for this peer. |
| Admin state | Administrative state of this peer. |
| Charged Units | Total number of charging units applying to this peer since system startup. |
| codec | Default voice coder rate of speech for this peer. |
| Connect Time | Accumulated connect time to the peer since system startup for both incoming and outgoing calls. |
| dest-pat | Destination pattern (telephone number) for this peer. |
| Expect factor | User-requested Expectation Factor of voice quality for calls via this peer. |
| fax-rate | Fax transmission rate configured for this peer. |
| Failed Calls | Number of failed call attempts to this peer since system startup. |
| group | Group number associated with this peer. |
| ICPIF | Configured ICPIF value for calls sent by a dial peer. |
| incall-number | Full E.164 telephone number to be used to identify the dial peer. |
| Last Disconnect Cause | Encoded network cause associated with the last call. This value is updated whenever a call is started or cleared and depends on the interface type and session protocol being used on this interface. |
| Last Disconnect Text | ASCII text describing the reason for the last call termination. |
| Last Setup Time | Value of the System Up Time when the last call to this peer was started. |
| Operation state | Operational state of this peer. |
| Permission | Configured permission level for this peer. |
| Poor QOV Trap | Whether poor-quality-of-voice trap messages have been enabled or disabled. |
| Refused Calls | Number of calls from this peer refused since system startup. |

*Table 5 Show-Dial-Peer-Voice Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| req-qos | Configured requested QoS for calls for this dial peer. |
| session target | Session target of this peer. |
| sess-proto | Session protocol to be used for Internet calls between local and remote router via the IP backbone. |
| Successful Calls | Number of completed calls to this peer. |
| tag | Unique dial-peer ID number. |
| VAD | Whether or not VAD is enabled for this dial peer. |

### Related Commands

**show call active voice**
**show call-history voice**
**show num-exp**
**show voice port**

# show dialplan incall number

To pair different voice ports and telephone numbers together for troubleshooting, use the **show dialplan incall number** privileged EXEC command.

**show dialplan incall** *slot-number/port* **number** *dial string*

### Syntax Description

| | |
|---|---|
| *slot-number* | Slot number in the router where the VIC is installed. Valid entries are from 0 to 2, depending on the VIC you have installed. |
| *port* | Voice port. Valid entries are 0 or 1. |
| *dial string* | Particular destination pattern (telephone number). |

### Command Mode

Privileged EXEC.

### Usage Guidelines

Occasionally, an incoming call cannot be matched to a dial peer in the dial-peer database. One reason this might occur is that the specified destination cannot be reached via the voice interface through which the incoming call came. Use the **show dialplan incall number** command as a troubleshooting method to resolve the call destination by pairing voice ports and telephone numbers together until there is a match.

### Example

The following example tests whether the telephone extension *57681* can be reached through voice port 0/1:

```
show dialplan incall 0/1 number 57681
```

### Related Commands

**show dialplan number**

# show dialplan number

To show which dial peer is reached when a particular telephone number is dialed, use the **show dial plan number** privileged EXEC command.

**show dial plan number** *dial string*

### Syntax Description

| | |
|---|---|
| *dial string* | Particular destination pattern (telephone number). |

### Command Mode

Privileged EXEC.

### Usage Guidelines

Use the **show dialplan number** command to test that the dial-plan configuration is valid and working as expected.

### Example

The following example displays the dial peer associated with the destination pattern of 54567:

```
show dialplan number 54567
```

### Related Commands

**show dialplan incall number**

# show num-exp

To show the number expansions configured, use the **show num-exp** privileged EXEC command.

**show num-exp** [*dialed- number*]

### Syntax Description

*dialed-number* Displays number expansion for the specified dialed number.

### Command Mode

Privileged EXEC.

### Usage Guidelines

Use the **show num-exp** privileged EXEC command to display all of the number expansions configured for this router. To display number expansion for only one number, specify that number by using the *dialed-number* argument.

### Sample Display

The following is sample output from the **show num-exp** command:

```
router# show num-exp
Dest Digit Pattern = '0...'     Translation = '14085550...'
Dest Digit Pattern = '1...'     Translation = '14085551...'
Dest Digit Pattern = '3..'      Translation = '140855503..'
Dest Digit Pattern = '4..'      Translation = '140855504..'
Dest Digit Pattern = '5..'      Translation = '140855505..'
Dest Digit Pattern = '6....'    Translation = '1408526....'
Dest Digit Pattern = '7....'    Translation = '1408527....'
Dest Digit Pattern = '8...'     Translation = '14085558...'
```

Table 5 explains the fields in the sample output.

***Table 6 Show-Dial-Peer-Voice Command Field Descriptions***

| Field | Description |
|---|---|
| Dest Digit Pattern | Index number identifying the destination telephone number digit pattern. |
| Translation | Expanded destination telephone number digit pattern. |

### Related Commands

**show call active voice**
**show call history voice**
**show dial-peer voice**
**show voice port**

# show voice dsp

To show the current status of all DSP voice channels, use the **show voice dsp** privileged EXEC command.

**show voice dsp**

### Syntax Description

This command has no arguments or keywords.

### Command Mode

Privileged EXEC.

### Usage Guidelines

This command also applies to Voice over Frame Relay, Voice over ATM, and Voice over HDLC on the Cisco MC3810.

### Sample Display

The following is sample output from the **show voice dsp** command:

```
router# show voice dsp
DSP#0: state IN SERVICE, 2 channels allocated
channel#0: voice port 1/0, codec G711 ulaw, state UP
channel#1: voice port 1/1, codec G711 ulaw, state UP
DSP#1: state IN SERVICE, 2 channels allocated
channel#0: voice port 2/0, codec G711 ulaw, state UP
channel#1: voice port 2/1, codec G711 ulaw, state UP
DSP#2: state RESET, 0 channels allocated
```

Table 7 explains the fields in the sample output.

***Table 7 Show Voice DSP Command Field Descriptions***

| Field | Description |
|---|---|
| DSP | Number of the DSP. |
| Channel | Number of the channel and its status. |

### Related Commands

**show dial-peer voice**
**show voice call summary**
**show voice port**

# show voice port

To display configuration information about a specific voice port, use the **show voice port** privileged EXEC command.

**show voice port** *slot-number/port*

### Syntax Description

| | |
|---|---|
| *slot-number* | Slot number in the router where the VIC is installed. Valid entries are from 0 to 2, depending on the slot where it has been installed. |
| *port* | Voice port. Valid entries are 0 or 1. |

### Command Mode

Privileged EXEC.

### Usage Guidelines

Use the **show voice port** privileged EXEC command to display configuration and VIC-specific information about a specific port.

## Sample Display

The following is sample output from the **show voice port** command for an E&M voice port:

```
router# show voice port 0/0
E&M Slot 0/0
 Type of VoicePort is E&M
 Operation State is unknown
 Administrative State is unknown
 The Interface Down Failure Cause is 0
 Alias is NULL
 Noise Regeneration is disabled
 Non Linear Processing is disabled
 Music On Hold Threshold is Set to 0 dBm
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is disabled
 Echo Cancel Coverage is set to 16ms
 Connection Mode is Normal
 Connection Number is
 Initial Time Out is set to 0 s
 Interdigit Time Out is set to 0 s
 Analog Info Follows:
 Region Tone is set for northamerica
 Currently processing none
 Maintenance Mode Set to None (not in mtc mode)
 Number of signaling protocol errors are 0

 Voice card specific Info Follows:
 Signal Type is wink-start
 Operation Type is 2-wire
 Impedance is set to 600r Ohm
 E&M Type is unknown
 Dial Type is dtmf
 In Seizure is inactive
 Out Seizure is inactive
 Digit Duration Timing is set to 0 ms
 InterDigit Duration Timing is set to 0 ms
 Pulse Rate Timing is set to 0 pulses/second
 InterDigit Pulse Duration Timing is set to 0 ms
 Clear Wait Duration Timing is set to 0 ms
 Wink Wait Duration Timing is set to 0 ms
 Wink Duration Timing is set to 0 ms
 Delay Start Timing is set to 0 ms
 Delay Duration Timing is set to 0 ms
```

The following is sample output from the **show voice port** command for an FXS voice port:

```
router# show voice port 0/0
Foreign Exchange Station 0/0 Slot is 0, Port is 0
 Type of VoicePort is FXS
 Operation State is DORMANT
 Administrative State is UP
 The Interface Down Failure Cause is 0
 Alias is NULL
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 Music On Hold Threshold is Set to 0 dBm
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is enabled
 Echo Cancel Coverage is set to 16ms
 Connection Mode is Normal
 Connection Number is
 Initial Time Out is set to 10 s
 Interdigit Time Out is set to 10 s
Analog Info Follows:
 Region Tone is set for northamerica
 Currently processing none
 Maintenance Mode Set to None (not in mtc mode)
 Number of signaling protocol errors are 0
 Voice card specific Info Follows:
 Signal Type is loopStart
 Ring Frequency is 25 Hz
 Hook Status is On Hook
 Ring Active Status is inactive
 Ring Ground Status is inactive
 Tip Ground Status is inactive
 Digit Duration Timing is set to 100 ms
 InterDigit Duration Timing is set to 100 ms
 Hook Flash Duration Timing is set to 600 ms
```

Table 8 explains the fields in the sample output.

***Table 8 Show-Voice-Port Command Field Descriptions***

| Field | Description |
|---|---|
| Administrative State | Administrative state of the voice port. |
| Alias | User-supplied alias for this voice port. |
| Clear Wait Duration Timing | Time of inactive seizure signal to declare call cleared. |
| Connection Mode | Connection mode of the interface. |
| Connection Number | Full E.164 telephone number used to establish a connection with the trunk or PLAR mode. |
| Currently Processing | Type of call currently being processed: none, voice, or fax. |
| Delay Duration Timing | Maximum delay signal duration for delay dial signaling. |
| Delay Start Timing | Timing of generation of delayed start signal from detection of incoming seizure. |
| Dial Type | Out-dialing type of the voice port. |
| Digit Duration Timing | DTMF Digit duration in milliseconds. |
| E&M Type | Type of E&M interface. |
| Echo Cancel Coverage | Echo cancel coverage for this port. |

*Table 8 Show-Voice-Port Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| Echo Cancellation | Whether or not echo cancellation is enabled for this port. |
| Hook Flash Duration Timing | Maximum length of hook flash signal. |
| Hook Status | Hook status of the FXO/FXS interface. |
| Impedance | Configured terminating impedance for the E&M interface. |
| In Gain | Amount of gain inserted at the receiver side of the interface. |
| In Seizure | Incoming seizure state of the E&M interface. |
| Initial Time Out | Amount of time the system waits for an initial input digit from the caller. |
| InterDigit Duration Timing | DTMF interdigit duration in milliseconds. |
| InterDigit Pulse Duration Timing | Pulse dialing interdigit timing in milliseconds. |
| Interdigit Time Out | Amount of time the system waits for a subsequent input digit from the caller. |
| Maintenance Mode | Maintenance mode of the voice port. |
| Music On Hold Threshold | Configured Music-On-Hold Threshold value for this interface. |
| Noise Regeneration | Whether or not background noise should be played to fill silent gaps if VAD is activated. |
| Number of signaling protocol errors | Number of signaling protocol errors. |
| Non-Linear Processing | Whether or not nonlinear processing is enabled for this port. |
| Operations State | Operation state of the port. |
| Operation Type | Operation of the E&M signal: two-wire or four-wire. |
| Out Attenuation | Amount of attenuation inserted at the transmit side of the interface. |
| Out Seizure | Outgoing seizure state of the E&M interface. |
| Port | Port number for this interface associated with the VIC. |
| Pulse Rate Timing | Pulse dialing rate in pulses per second (pps). |
| Regional Tone | Configured regional tone for this interface. |
| Ring Active Status | Ring active indication. |
| Ring Frequency | Configured ring frequency for this interface. |
| Ring Ground Status | Ring ground indication. |
| Signal Type | Type of signaling for a voice port: loop-start, ground-start, wink-start, immediate, and delay-dial. |
| Slot | Slot used in the VIC for this port. |
| Tip Ground Status | Tip ground indication. |
| Type of VoicePort | Type of voice port: FXO, FXS, and E&M. |
| The Interface Down Failure Cause | Text string describing why the interface is down. |

*Table 8 Show-Voice-Port Command Field Descriptions (Continued)*

| Field | Description |
|---|---|
| Wink Duration Timing | Maximum wink duration for wink start signaling. |
| Wink Wait Duration Timing | Maximum wink wait duration for wink start signaling. |

### Related Commands

**show call active voice**
**show call history voice**
**show dial-peer voice**
**show num-exp**

# shutdown (dial-peer configuration)

To change the administrative state of the selected dial peer from up to down, use the **shutdown** dial-peer configuration command. Use the **no** form of this command to change the administrative state of this dial peer from down to up.

**shutdown**
**no shutdown**

### Syntax Description

This command has no arguments or keywords.

### Default

No state is predefined.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

When a dial peer is shut down, you cannot initiate calls to that peer. This command applies to both VoIP and POTS peers.

### Example

The following example changes the administrative state of voice telephony dial peer 10 to down:

```
configure terminal
 dial-peer voice 10 pots
 shutdown
```

# shutdown (voice-port configuration)

To take the voice ports for a specific VIC offline, use the **shutdown** voice-port configuration command. Use the **no** form of this command to put the ports back in service.

**shutdown**
**no shutdown**

### Syntax Description

This command has no arguments or keywords.

### Default

Enabled.

### Command Mode

Voice-port configuration.

### Usage Guidelines

When you enter the **shutdown** command, all ports on the VIC are disabled, and there is dead silence on the telephone connected to the interface. When you enter the **no shutdown** command, all ports on the VIC are enabled.

### Example

The following example takes voice port 1/0 offline:

```
configure terminal
 voice port 1/0
 shutdown
```

**Note** The preceding configuration example first shuts down voice port 1/0 and then voice port 1/1.

# signal

To specify the type of signaling for a voice port, use the **signal** voice-port configuration command. Use the **no** form of this command to restore the default value for this command.

**signal** {**loop-start** | **ground-start** | **wink-start** | **immediate** | **delay-dial**}
**no signal**

### Syntax Description

| | |
|---|---|
| **loop-start** | Loop Start signaling. Used for FXO and FXS interfaces. With Loop Start signaling, only one side of a connection can hang up. This is the default setting for FXO and FXS voice ports. |
| **ground-start** | Ground Start signaling. Used for FXO and FXS interfaces. Ground Start allows both sides of a connection to place a call and to hang up. |
| wink-start | Calling side seizes the line by going off-hook on its E lead and then waits for a short off-hook “wink” indication on its M lead from the called side before sending address information as DTMF digits. Used for E&M tie trunk interfaces. This is the default setting for E&M voice ports. |
| immediate | Calling side seizes the line by going off-hook on its E lead and sends address information as DTMF digits. Used for E&M tie trunk interfaces. |
| delay-dial | Calling side seizes the line by going off-hook on its E lead. After a timing interval, the calling side looks at the supervision from the called side. If the supervision is on-hook, the calling side starts sending information as DTMF digits; otherwise, the calling side waits until the called side goes on-hook and then starts sending address information. Used for E&M tie trunk interfaces. |

### Default

**loop-start** for FXO and FXS interfaces.

**wink-start** for E&M interfaces.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Configuring the **signal** command for an FXS or FXO voice port changes the signal value for both voice ports on a VIC.

**Note** If you change the signal type for an FXO voice port, you need to move the appropriate jumper in the VIC.

Configuring this command for an E&M voice port changes only the signal value for the selected voice port. In either case, the voice port must be shut down and then activated before the configured values take effect.

Some PBXs miss initial digits if the E&M voice port is configured for **immediate** signaling. If this occurs, use **delay-dial** signaling instead. Some devices (not Cisco devices) have a limited number of DTMF receivers. This type of equipment must delay the calling side until a DTMF receiver is available.

### Example

The following example configures **ground-start** signaling, which means that both sides of a connection can place a call and hang up, as the signaling type for a voice port:

```
configure terminal
 voice port 1/1
 signal ground-start
```

# snmp enable peer-trap poor-qov

To generate poor-quality-of-voice notification for applicable calls associated with VoIP dial peers, use the **snmp enable peer-trap poor-qov** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**snmp enable peer-trap poor-qov**
**no snmp enable peer-trap poor-qov**

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **snmp enable peer-trap poor qov** command to generate poor-quality-of-voice notifications for applicable calls associated with this dial peer. If you have an SNMP manager that uses SNMP messages when voice quality drops, you might want to enable this command. Otherwise, you should disable this command to reduce unnecessary network traffic.

This command only applies to VoIP peers.

### Example

The following example enables poor-quality-of-voice notifications for calls associated with VoIP dial peer 10:

```
dial-peer voice 10 voip
 snmp enable peer-trap poor-qov
```

### Related Commands

**snmp-server enable traps voice poor-qov**
**snmp trap link-status**

# snmp-server enable traps

To enable the router to send SNMP traps, use the **snmp-server enable traps** global configuration command. Use the **no** form of this command to disable SNMP traps.

**snmp-server enable traps** [*trap-type*] [*trap-option*]
**no snmp-server enable traps** [*trap-type*] [*trap-option*]

## Defaults

No traps are enabled.

Some trap types cannot be controlled with this command. These traps are either always enabled or enabled by some other means. For example, the linkUpDown messages are disabled by the **no snmp trap link-status** command.

If you enter this command with no keywords, the default is to enable all trap types.

## Command Mode

Global configuration.

## Usage Guidelines

This command is useful for disabling traps that are generating a large amount of uninteresting or useless noise.

If you do not enter an **snmp-server enable traps** command, no traps controlled by this command are sent. To configure the router to send these SNMP traps, you must enter at least one **snmp-server enable traps** command. If you enter the command with no keywords, all trap types are enabled. If you enter the command with a keyword, only the trap type related to that keyword is enabled. To enable multiple types of traps, you must issue a separate **snmp-server enable traps** command for each trap type and option.

The **snmp-server enable traps** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP traps. In order to send traps, you must configure at least one **snmp-server host** command.

For a host to receive a trap controlled by this command, both the **snmp-server enable traps** command and the **snmp-server host** command for that host must be enabled. If the trap type is not controlled by this command, just the appropriate **snmp-server host** command must be enabled.

The trap types used in this command all have an associated MIB object that allows them to be globally enabled or disabled. Not all of the trap types available in the **snmp-server host** command have notificationEnable MIB objects, so some of these cannot be controlled using the **snmp-server enable traps** command.

## Examples

The following example enables the router to send SNMP poor-quality-of-voice traps:

```
configure terminal
 snmp-server enable trap voice poor-qov
```

The following example enables the router to send all traps to the host myhost.cisco.com using the community string *public*:

```
snmp-server enable traps
snmp-server host myhost.cisco.com public
```

The following example enables the router to send Frame Relay and environmental monitor traps to the host myhost.cisco.com using the community string *public*:

```
snmp-server enable traps frame-relay
snmp-server enable traps envmon temperature
snmp-server host myhost.cisco.com public
```

The following example does not send traps to any host. The BGP traps are enabled for all hosts, but the only traps enabled to be sent to a host are ISDN traps.

```
snmp-server enable traps bgp
snmp-server host bob public isdn
```

### Related Commands

**snmp enable peer-trap peer-qov**
**snmp-server host**
**snmp-server trap-source**
**snmp trap illegal-address**
**snmp trap link-status**

# snmp trap link-status

To enable SNMP trap messages to be generated when this voice port is brought up or down, use the **snmp trap link-status** voice-port configuration command. Use the **no** form of this command to disable this feature.

**snmp trap link-status**
**no snmp trap link-status**

### Syntax Description

This command contains no arguments or keywords.

### Default

Enabled.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **snmp trap link-status** command to enable SNMP trap messages (linkup and linkdown) to be generated whenever this voice port is brought online or offline.

If you are managing the equipment with an SNMP manager (such as Maestro), enable this command. Enabling **link-status** messages allows the SNMP manager to learn of a status change without polling the equipment. If you are not using an SNMP manager, disable this command to avoid unnecessary network traffic.

### Example

The following example enables SNMP trap messages for voice port 1/0:

```
voice port 1/0
 snmp trap link-status
```

### Related Commands

**snmp enable peer-trap poor-qov**
snmp-server enable traps poor-qov

# timeouts initial

To configure the initial digit timeout value for a specified voice port, use the **timeouts initial** voice-port configuration command. Use the **no** form of this command to restore the default value for this command.

**timeouts initial** *seconds*
**no timeouts initial** *seconds*

### Syntax Description

| | |
|---|---|
| *seconds* | Initial timeout duration in seconds. Valid entries are any integer from 0 to 120. |

### Default

10 seconds.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **timeouts initial** command to specify the number of seconds the system waits for the caller to enter the first digit of the dialed digits. The timeouts initial timer is activated when the call is accepted and is deactivated when the caller enters the first digit. If the configured timeout value is exceeded, the caller is notified through the appropriate tone, and the call is terminated.

To disable the timeouts initial timer, set the *seconds* value to 0.

### Example

The following example sets the initial digit timeout value to 15 seconds:

```
voice port 0/0
 timeouts initial 15
```

### Related Commands

**timeouts interdigit**
**timing**

# timeouts interdigit

To configure the interdigit timeout value for a specified voice port, use the **timeouts interdigit** voice-port configuration command. Use the **no** form of this command to restore the default value for this command.

**timeouts interdigit** *seconds*
**no timeouts interdigit** *seconds*

### Syntax Description

| | |
|---|---|
| *seconds* | Interdigit timeout duration in seconds. Valid entries are any integer from 0 to 120. |

### Default

10 seconds.

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **timeouts interdigit** command to specify the number of seconds the system waits (after the caller has entered the initial digit) for the caller to enter a subsequent digit of the dialed digits. The timeouts interdigit timer is activated when the caller enters a digit and is restarted each time the caller enters another digit until the destination address is identified. If the configured timeout value is exceeded before the destination address is identified, the caller is notified through the appropriate tone, and the call is terminated.

To disable the timeouts interdigit timer, set the *seconds* value to 0.

### Example

The following example sets the interdigit timeout value to 15 seconds:

```
voice port 0/0
 timeouts interdigit 15
```

### Related Commands

**timeouts initial**
**timing**

# timing

To specify timing parameters (other than those defined by the **timeouts** commands) for a specified voice port, use the **timing** voice-port configuration command. Use the **no** form of this command to reset the default value for this command.

**timing** *timing-value*
**no timing** *timing-value*

### Syntax Description

*timing-value* One of the keyword/argument pairs listed in Table 9.

***Table 9 Timing Keywords/Arguments, Descriptions, and Valid Entries***

| Keyword/Argument | Argument Description | Valid Entries |
|---|---|---|
| **clear-wait** *milliseconds* | The minimum amount of time, in milliseconds, between the inactive seizure signal and the call being cleared | Numbers from 200 to 2000 |
| **delay-duration** *milliseconds* | The delay signal duration for delay dial signaling, in milliseconds | Numbers from 100 to 5000 |
| **delay-start** *milliseconds* | The minimum delay time, in milliseconds, from outgoing seizure to outdial address | Numbers from 20 to 2000 |
| **dial-pulse min-delay** *milliseconds* | The time, in milliseconds, between the generation of wink-like pulses | Numbers from 0 to 5000 |
| **digit** *milliseconds* | The DTMF digit signal duration, in milliseconds | Numbers from 50 to 100 |
| **inter-digit** *milliseconds* | The DTMF inter-digit duration, in milliseconds | Numbers from 50 to 500 |
| **pulse** *pulses per second* | The pulse dialing rate, in pulses per second | Numbers from 10 to 20 |
| **pulse-inter-digit** *milliseconds* | The pulse dialing inter-digit timing, in milliseconds | Numbers from 100 to 1000 |

*Table 9 Timing Keywords/Arguments, Descriptions, and Valid Entries*

| | | |
|---|---|---|
| **wink-duration** *milliseconds* | The maximum wink signal duration, in milliseconds, for a wink start signal | Numbers from 100 to 400 |
| **wink-wait** *milliseconds* | The maximum wink-wait duration, in milliseconds, for a wink start signal | Numbers from 100 to 5000 |

## Default

The default values for the **timing** keywords/arguments are listed in Table 10.

*Table 10 Timing Keywords/Arguments Default Values*

| Keyword/Argument | Default Value |
|---|---|
| **clear-wait** *milliseconds* | 400 ms |
| **delay-duration** *milliseconds* | 2000 ms |
| **delay-start** *milliseconds* | 300 ms |
| **dial-pulse min-delay** *milliseconds* | 140 ms |
| **digit** *milliseconds* | 100 ms |
| **inter-digit** *milliseconds* | 100 ms |
| **pulse** *pulses per second* | 20 pps |
| **pulse-inter-digit** *milliseconds* | 500 ms |
| **wink-duration** *milliseconds* | 200 ms |
| **wink-wait** *milliseconds* | 200 ms |

## Command Mode

Voice-port configuration.

## Usage Guidelines

Use the **timing** command to specify timing parameters other than those defined by the **timeouts** commands.

Use the **timing** command with the **dial-pulse min-delay** keyword with PBXs requiring a wink-like pulse, even though they have been configured for delay-dial signaling. If the value for this keyword is set to 0, the router does not generate this wink-like pulse.

Table 11 lists the call signal directions for the **timing** keyword/argument pairs.

*Table 11 Timing Keywords/Arguments Call Signal Directions*

| Timing Keyword/Argument | Call Signal Direction |
|---|---|
| **clear-wait** *milliseconds* | Not applicable |
| **delay-duration** *milliseconds* | Out |
| **delay-start** *milliseconds* | Out |

*Table 11 Timing Keywords/Arguments Call Signal Directions*

| Timing Keyword/Argument | Call Signal Direction |
|---|---|
| **dial-pulse min-delay** *milliseconds* | In |
| **digit** *milliseconds* | Out |
| **inter-digit** *milliseconds* | Out |
| **pulse** *pulses per second* | Out |
| **pulse-inter-digit** *milliseconds* | Out |
| **wink-duration** *milliseconds* | Out |
| **wink-wait** *milliseconds* | Out |

### Example

The following example configures the clear-wait duration to 300 milliseconds:

```
voice port 0/0
 timing clear-wait 300
```

### Related Commands

**timeouts initial**
**timeouts interdigit**

# type

To specify the E&M interface type, use the **type** voice-port configuration command. Use the **no** form of this command to reset the default value for this command.

**type** {**1** | **2** | **3** | **5**}
**no type**

### Syntax Description

| | |
|---|---|
| **1** | For the following lead configuration:<br>E—Output, relay to ground.<br>M—Input, referenced to ground. |
| **2** | For the following lead configuration:<br>E—Output, relay to SG.<br>M—Input, referenced to ground.<br>SB—Feed for M, connected to –48V.<br>SG—Return for E, galvanically isolated from ground. |
| 3 | For the following lead configuration:<br>E—Output, relay to ground.<br>M—Input, referenced to ground.<br>SB—Connected to –48V.<br>SG—Connected to ground. |
| 5 | For the following lead configuration:<br>E—Output, relay to ground.<br>M—Input, referenced to –48V. |

### Default

1

### Command Mode

Voice-port configuration.

### Usage Guidelines

Use the **type** command to specify the E&M interface for a particular voice port. With **1**, the tie-line equipment generates the E-signal to the PBX by grounding the E-lead. The tie-line equipment detects the M-signal by detecting current flow to ground. If you select **1**, a common ground must exist between the line equipment and the PBX.

With **2**, the interface requires no common ground between the equipment, thereby avoiding ground loop noise problems. The tie-line equipment generates the E-signal to the PBX by connecting it to SG. The M-signal is detected by the PBX connecting it to SB. Although Type 2 interfaces do not require a common ground, they do have the tendency to inject noise into the audio paths because they are asymmetrical with respect to the current flow between devices.

With **3**, the interface operates the same as type 1 interfaces with respect to the E-signal. However, the M-signal is detected by the PBX connecting it to SB on assertion and alternately connecting it to SG during inactivity. If you select **3**, a common ground must be shared between equipment.

With **5**, the type 5 line equipment generates the E-signal to the PBX by grounding the E-lead. The PBX detects M-signal by grounding the M-lead. A type 5 interface is quasi-symmetrical in that, while the line is up, current flow is more or less equal between the PBX and the line equipment, but noise injection is a problem.

### Example

The following example selects type 3 as the interface type for your voice port:

```
voice port 0/0
 type 3
```

# vad

To enable voice activity detection (VAD) for the calls using this dial peer, use the **vad** dial-peer configuration command. Use the **no** form of this command to disable this feature.

**vad**
**no vad**

### Syntax Description

This command has no arguments or keywords.

### Default

Enabled.

### Command Mode

Dial-peer configuration.

### Usage Guidelines

Use the **vad** command to enable VAD. With VAD, silence is not transmitted over the network, only audible speech. If you enable VAD, the sound quality is slightly degraded, but the connection monopolizes much less bandwidth. If you use the **no** form of this command, VAD is disabled, and voice data is continuously transmitted to the IP backbone.

This command only applies to VoIP peers.

### Example

The following example enables VAD:

```
dial-peer voice 10 voip
 vad
```

### Related Commands

**comfort-noise**

# voice-port

To enter the voice port configuration mode, use the **voice-port** global configuration command.

**voice-port** *slot-number/port*

### Syntax Description

| | |
|---|---|
| *slot-number* | Slot number in the router where the VIC is installed. Valid entries are from 0 to 2, depending on the slot where it has been installed. |
| *port* | Voice port. Valid entries are 0 or 1. |

### Default

No voice-port mode is configured.

### Command Mode

Global configuration.

### Usage Guidelines

Use the **voice-port** global configuration command to switch to the voice port configuration mode from the global configuration mode. Use the **exit** command to exit the voice port configuration mode and return to the global configuration mode.

### Example

The following example accesses the voice port configuration mode for a VIC installed in port 0, slot 0:

```
configure terminal
 voice port 0/0
```

### Related Commands

**dial-peer**

# 5

# VoIP Debug Commands

This chapter documents debug commands that are new or specific to the Cisco 1751 router. All other commands used with this feature are documented in the *Debug Command Reference* chapter for the Cisco IOS Release12.1T.

- **debug voip ccapi error**
- **debug voip ccapi inout**
- **debug vpm all**
- **debug vpm dsp**
- **debug vpm error**
- **debug vpm port**
- **debug vpm signal**
- **debug vpm spi**
- **debug vtsp all**
- **debug vtsp dsp**
- **debug vtsp error**
- **debug vtsp port**
- **debug vtsp session**
- **debug vtsp stats**
- **debug vtsp tone**
- **debug vtsp vofr subframe**

## Using Debug Commands

Debug commands are provided for most of the configurations in this document. You can use the debug commands to troubleshoot any configuration problems that you might be having on your network. Debug commands provide extensive, informative displays to help you interpret any possible problems.

Table 5-1 contains important information about debug commands.

**Caution** Debugging is assigned a high priority in your router CPU process, and it can render your router unusable. For this reason, use debug commands only to troubleshoot specific problems. The best time to use debug commands is during periods of low network traffic and few users to decrease the likelihood that the debug command processing overhead affects network users.

***Table 1 Important Information About Debug Commands***

| About | Information |
|---|---|
| Additional documentation | You can find additional information and documentation about the debug commands in the *Debug Command Reference* document on the Cisco IOS software documentation CD-ROM that came with your router.<br><br>If you are not sure where to find this document on the CD-ROM, use the Search function in the Verity Mosaic browser that comes with the CD-ROM. |
| Disabling debugging | To turn off any debugging, enter the **undebug all** command. |
| Telnet sessions | If you want to use debug command during a telnet session with your router, you must first enter the **terminal monitor** command. |

# debug voip ccapi error

Use the **debug voip ccapi error** EXEC command to trace error logs in the call control application programming interface (API). Use the **no** form of this command to disable debugging output.

**[no] debug voip ccapi error**

## Usage Guidelines

The **debug voip ccapi error** EXEC command traces the error logs in the call control API. When there are insufficient resources, error logs are generated during normal call processing. They are also generated when there are problems in the underlying network-specific code, the higher call session application, or the call control API itself.

This debug command shows error events or unexpected behavior in system software. In most cases, no events are generated.

# debug voip ccapi inout

Use the **debug voip ccapi inout** EXEC command to trace the execution path through the call control application programming interface (API). Use the **no** form of this command to disable debugging output.

**[no] debug voip ccapi inout**

### Usage Guidelines

The **debug voip ccapi inout** EXEC command traces the execution path through the call control API, which serves as the interface between the call session application and the underlying network-specific software. You can use the output from this command to understand how calls are being handled by the router.

This command shows how a call flows through the system. Using this debug level, you can see the call setup and teardown operations performed on both the telephony and network call legs.

### Sample Display

The following output shows the call setup indicated and accepted by the router:

```
router# debug voip ccapi inout
cc_api_call_setup_ind (vdbPtr=0x60BFB530, callInfo={called=, calling=, fdest=0},
callID=0x60BFAEB8)
cc_process_call_setup_ind (event=0x60B68478)
sess_appl: ev(14), cid(1), disp(0)
ccCallSetContext (callID=0x1, context=0x60A7B094)
ccCallSetPeer (callID=0x1, peer=0x60C0A868, voice_peer_tag=2, encapType=1,
dest-pat=14085231001, answer=)
ccCallSetupAck (callID=0x1)
```

The following output shows the caller entering DTMF digits until a dial-peer is matched:

```
cc_api_call_digit (vdbPtr=0x60BFB530, callID=0x1, digit=4, mode=0)
sess_appl: ev(8), cid(1), disp(0)
ssa: cid(1)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
cc_api_call_digit (vdbPtr=0x60BFB530, callID=0x1, digit=1, mode=0)
sess_appl: ev(8), cid(1), disp(0)
ssa: cid(1)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
cc_api_call_digit (vdbPtr=0x60BFB530, callID=0x1, digit=0, mode=0)
sess_appl: ev(8), cid(1), disp(0)
ssa: cid(1)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
cc_api_call_digit (vdbPtr=0x60BFB530, callID=0x1, digit=0, mode=0)
sess_appl: ev(8), cid(1), disp(0)
ssa: cid(1)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
cc_api_call_digit (vdbPtr=0x60BFB530, callID=0x1, digit=1, mode=0)
sess_appl: ev(8), cid(1), disp(0)
ssa: cid(1)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
ccCallProceeding (callID=0x1, prog_ind=0x0)
ssaSetupPeer cid(1), destPat(14085241001), matched(8), prefix(), peer(60C0E710)
```

The following output shows the call setup over the IP network to the remote router:

```
ccCallSetupRequest (peer=0x60C0E710, dest=, params=0x60A7B0A8 mode=0, *callID=0x60B6C110)
ccIFCallSetupRequest: (vdbPtr=0x60B6C5D4, dest=, callParams={called=14085241001,
calling=14085231001, fdest=0, voice_peer_tag=104}, mode=0x0)
ccCallSetContext (callID=0x2, context=0x60A7B2A8)
```

The following output shows the called party is alerted, a codec is negotiated, and voice path is cut through:

```
cc_api_call_alert(vdbPtr=0x60B6C5D4, callID=0x2, prog_ind=0x8, sig_ind=0x1)
sess_appl: ev(6), cid(2), disp(0)
ssa: cid(2)st(1)oldst(0)cfid(-1)csize(0)in(0)fDest(0)-cid2(1)st2(1)oldst2(0)
ccCallAlert (callID=0x1, prog_ind=0x8, sig_ind=0x1)
ccConferenceCreate (confID=0x60B6C150, callID1=0x1, callID2=0x2, tag=0x0)
cc_api_bridge_done (confID=0x1, srcIF=0x60B6C5D4, srcCallID=0x2, dstCallID=0x1,
disposition=0, tag=0x0)
cc_api_bridge_done (confID=0x1, srcIF=0x60BFB530, srcCallID=0x1, dstCallID=0x2,
disposition=0, tag=0x0)
cc_api_caps_ind (dstVdbPtr=0x60B6C5D4, dstCallId=0x2,srcCallId=0x1, caps={codec=0x7,
fax_rate=0x7F, vad=0x3})
cc_api_caps_ind (dstVdbPtr=0x60BFB530, dstCallId=0x1,srcCallId=0x2, caps={codec=0x4,
fax_rate=0x2, vad=0x2})
cc_api_caps_ack (dstVdbPtr=0x60BFB530, dstCallId=0x1,srcCallId=0x2, caps={codec=0x4,
fax_rate=0x2, vad=0x2})
cc_api_caps_ack (dstVdbPtr=0x60B6C5D4, dstCallId=0x2,srcCallId=0x1, caps={codec=0x4,
fax_rate=0x2, vad=0x2})
sess_appl: ev(17), cid(1), disp(0)
ssa: cid(1)st(3)oldst(0)cfid(1)csize(0)in(1)fDest(0)-cid2(2)st2(3)oldst2(1)
```

The following output shows that the call is connected and voice is active:

```
cc_api_call_connected(vdbPtr=0x60B6C5D4, callID=0x2)
sess_appl: ev(7), cid(2), disp(0)
ssa: cid(2)st(4)oldst(1)cfid(1)csize(0)in(0)fDest(0)-cid2(1)st2(4)oldst2(3)
ccCallConnect (callID=0x1)
```

The following output shows how the system processes voice statistics and monitors voice quality during the call:

```
ccapi_request_rt_packet_stats (requestorIF=0x60B6C5D4, requestorCID=0x2,
      requestedCID=0x1, tag=0x60A7C598)
cc_api_request_rt_packet_stats_done (requestedIF=0x60BFB530, requestedCID=0x1,
      tag=0x60A7A4C4)
ccapi_request_rt_packet_stats (requestorIF=0x60B6C5D4, requestorCID=0x2,
      requestedCID=0x1, tag=0x60A7C598)
cc_api_request_rt_packet_stats_done (requestedIF=0x60BFB530, requestedCID=0x1,
      tag=0x60C1FE54)
ccapi_request_rt_packet_stats (requestorIF=0x60B6C5D4, requestorCID=0x2,
      requestedCID=0x1, tag=0x60A7C598)
cc_api_request_rt_packet_stats_done (requestedIF=0x60BFB530, requestedCID=0x1,
      tag=0x60A7A5F4)
ccapi_request_rt_packet_stats (requestorIF=0x60B6C5D4, requestorCID=0x2,
      requestedCID=0x1, tag=0x60A7C598)
cc_api_request_rt_packet_stats_done (requestedIF=0x60BFB530, requestedCID=0x1,
      tag=0x60A7A6D8)
ccapi_request_rt_packet_stats (requestorIF=0x60B6C5D4, requestorCID=0x2,
      requestedCID=0x1, tag=0x60A7C598)
cc_api_request_rt_packet_stats_done (requestedIF=0x60BFB530, requestedCID=0x1,
      tag=0x60A7ACBC)
```

The following output shows that disconnection is generated from the calling party and that call legs are torn down and disconnected:

```
cc_api_call_disconnected(vdbPtr=0x60BFB530, callID=0x1, cause=0x10)
sess_appl: ev(9), cid(1), disp(0)
ssa: cid(1)st(5)oldst(3)cfid(1)csize(0)in(1)fDest(0)-cid2(2)st2(5)oldst2(4)
ccConferenceDestroy (confID=0x1, tag=0x0)
cc_api_bridge_done (confID=0x1, srcIF=0x60B6C5D4, srcCallID=0x2, dstCallID=0x1,
disposition=0 tag=0x0)
cc_api_bridge_done (confID=0x1, srcIF=0x60BFB530, srcCallID=0x1, dstCallID=0x2,
disposition=0 tag=0x0)
sess_appl: ev(18), cid(1), disp(0)
ssa: cid(1)st(6)oldst(5)cfid(-1)csize(0)in(1)fDest(0)-cid2(2)st2(6)oldst2(4)
ccCallDisconnect (callID=0x1, cause=0x10 tag=0x0)
ccCallDisconnect (callID=0x2, cause=0x10 tag=0x0)
cc_api_call_disconnect_done(vdbPtr=0x60B6C5D4, callID=0x2, disp=0, tag=0x0)
sess_appl: ev(10), cid(2), disp(0)
ssa: cid(2)st(7)oldst(4)cfid(-1)csize(0)in(0)fDest(0)-cid2(1)st2(7)oldst2(6)
cc_api_call_disconnect_done(vdbPtr=0x60BFB530, callID=0x1, disp=0, tag=0x0)
sess_appl: ev(10), cid(1), disp(0)
ssa: cid(1)st(7)oldst(6)cfid(-1)csize(1)in(1)fDest(0)
```

# debug vpm all

Use the **debug vpm all** EXEC command to enable debugging on all virtual voice-port module (VPM) areas. Use the **no** form of this command to disable debugging output.

**[no] debug vpm all**

### Usage Guidelines

The **debug vpm all** EXEC command enables all of the **debug vpm** commands: **debug vpm spi**, **debug vpm signal**, and **debug vpm dsp**. For more information or sample output, refer to the individual commands in this chapter.

# debug vpm dsp

Use the **debug vpm dsp** EXEC command to show messages from the digital signal processor (DSP) on the virtual voice-port module (VPM) to the router. Use the **no** form of this command to disable debugging output.

**[no] debug vpm dsp**

### Usage Guidelines

The **debug vpm dsp** command shows messages from the DSP on the VPM to the router; this command can be useful if you suspect that the VPM is not functional. It is a simple way to check if the VPM is responding to off-hook indications and to evaluate timing for signaling messages from the interface.

### Sample Display

The following output shows the DSP timestamp and the router timestamp for each event and, for SIG_STATUS, the state value shows the state of the ABCD bits in the signaling message. This sample shows a call coming in on a foreign exchange office (FXO) interface.

The router waits for ringing to terminate before accepting the call. State=0x0 indicates ringing; state 0x4 indicates not ringing:

```
router# debug vpm dsp
ssm_dsp_message: SEND/RESP_SIG_STATUS: state=0x0 timestamp=58172 systime=40024
ssm_dsp_message: SEND/RESP_SIG_STATUS: state=0x4 timestamp=59472 systime=40154
ssm_dsp_message: SEND/RESP_SIG_STATUS: state=0x4 timestamp=59589 systime=40166
```

The following output shows the digits collected:

```
vcsm_dsp_message: MSG_TX_DTMF_DIGIT: digit=4
vcsm_dsp_message: MSG_TX_DTMF_DIGIT: digit=1
vcsm_dsp_message: MSG_TX_DTMF_DIGIT: digit=0
vcsm_dsp_message: MSG_TX_DTMF_DIGIT: digit=0
vcsm_dsp_message: MSG_TX_DTMF_DIGIT: digit=0
```

This shows the disconnect indication and the final call statistics reported by the DSP (which are then populated in the call history table):

```
ssm_dsp_message: SEND/RESP_SIG_STATUS: state=0xC timestamp=21214 systime=42882
vcsm_dsp_message: MSG_TX_GET_TX_STAT: num_tx_pkts=1019 num_signaling_pkts=0
num_comfort_noise_pkts=0 transmit_durtation=24150 voice_transmit_duration=20380
fax_transmit_duration=0
```

# debug vpm error

Use the **debug vpm error** command to enable DSP error tracing in voice port modules (VPMs). Use the **no** form of this command to disable DSP error tracing.

[**no**] **debug vpm error**

### Usage Guidelines

Execution of **no debug all** will turn off all port level debugging. You should turn off all debugging and then enter the debug commands you are interested in one by one. This will help avoid confusion about which ports you are actually debugging.

# debug vpm port

Use the **debug vpm port** EXEC command to limit the debug output to a particular port. Use the **no** form of this command to disable debugging output.

[**no**] **debug vpm port** *slot-number*/*port*

### Syntax Description

| | |
|---|---|
| *slot-number* | Slot number in the router where the VIC is installed. Valid entries are from 0 to 2, depending on the slot where it has been installed. |
| *port* | Voice port. Valid entries are 0 or 1. |

### Usage Guidelines

Use the **debug vpm port** command to limit the debug output to a particular port. The debug output can be quite voluminous for a single port. A six-port chassis might create problems. Use this debug command with any or all of the other debug modes.

### Examples

The following example shows **debug vpm dsp** messages only for port 0/0:

```
debug vpm dsp
debug vpm port 0/0
```

The following example shows the **debug vpm signal** messages only for ports 0/0 and 0/1:

```
debug vpm signal
debug vpm port 0/0
debug vpm port 0/1
```

The following example shows how to turn off debugging on a port:

```
no debug vpm port 0/0
```

The following example shows no output because port level debugs work in conjunction with other levels:

```
debug vpm port 0/0
```

Execution of **no debug all** turns off all port level debugging. It is usually a good idea to turn off all debugging and then, one by one, to enter the debug commands you are interested in. This helps to avoid confusion about which ports you are actually debugging.

# debug vpm signal

Use the **debug vpm signal** EXEC command to collect debug information only for signaling events. Use the **no** form of this command to disable debugging output.

[**no**] **debug vpm signal**

### Usage Guidelines

The **debug vpm signal** EXEC command collects debug information only for signaling events. This command can also be useful in resolving problems with signaling to a PBX.

### Sample Display

The following output shows that a ring is detected and that the router waits for the ringing to stop before accepting the call:

```
router# debug vpm signal
ssm_process_event: [1/0, 0.2, 15] fxols_onhook_ringing
ssm_process_event: [1/0, 0.7, 19] fxols_ringing_not
ssm_process_event: [1/0, 0.3, 6]
ssm_process_event: [1/0, 0.3, 19] fxols_offhook_clear
```

The following output shows that the call is connected:

```
ssm_process_event: [1/0, 0.3, 4] fxols_offhook_proc
ssm_process_event: [1/0, 0.3, 8] fxols_proc_voice
ssm_process_event: [1/0, 0.3, 5] fxols_offhook_connect
```

The following output confirms a disconnect from the switch and release with higher layer code:

```
ssm_process_event: [1/0, 0.4, 27] fxols_offhook_disc
ssm_process_event: [1/0, 0.4, 33] fxols_disc_confirm
ssm_process_event: [1/0, 0.4, 3] fxols_offhook_release
```

# debug vpm spi

Use the **debug vpm spi** EXEC command to trace how the virtual voice-port module (VPM) serial peripheral interface (SPI) interfaces with the call control application programming interface (API). Use the **no** form of this command to disable debugging output.

[**no**] **debug vpm spi**

### Usage Guidelines

The **debug vpm spi** EXEC command traces how the virtual voice-port module SPI interfaces with the call control API. This debug command displays information about how each network indication and application request is handled.

This debug level shows the internal workings of the voice telephony call state machine.

### Sample Display

The following output shows that the call is accepted and presented to a higher layer code:

```
router# debug vpm spi
sp_set_sig_state: [1/0] packet_len=14 channel_id=129 packet_id=39 state=0xC timestamp=0x0
vcsm_process_event: [1/0, 0.5, 1] act_up_setup_ind
```

The following output shows that the higher layer code accepts the call, requests addressing information, and starts DTMF and dial-pulse collection. This also shows that the digit timer is started.

```
vcsm_process_event: [1/0, 0.6, 11] act_setup_ind_ack
dsp_voice_mode: [1/0 packet_len=22 channel_id=1 packet_id=73 coding_type=1
voice_field_size=160 VAD_flag=0 echo_length=128 comfort_noise=1 fax_detect=1
dsp_dtmf_mode: [1/0] packet_len=12 channel_id=1 packet_id=65 dtmf_or_mf=0
dsp_CP_tone_on: [1/0] packet_len=32 channel_id=1 packet_id=72 tone_id=3 n_freq=2
freq_of_first=350 freq_of_second=440 amp_of_first=4000 amp_of_second=4000 direction=1
on_time_first=65535 off_time_first=0 on_time_second=65535 off_time_second=0
dsp_digit_collect_on: [1/0] packet_len=22 channel_id=129 packet_id=35 min_inter_delay=550
max_inter_delay=3200 mim_make_time=18 max_make_time=75 min_brake_time=18
max_brake_time=75
vcsm_timer: 46653
```

The following output shows the collection of digits one by one until the higher level code indicates it has enough. The input timer is restarted with each digit, and the device waits in idle mode for connection to proceed.

```
vcsm_process_event: [1/0, 0.7, 25] act_dcollect_digit
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
vcsm_timer: 47055
vcsm_process_event: [1/0, 0.7, 25] act_dcollect_digit
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
vcsm_timer: 47079
vcsm_process_event: [1/0, 0.7, 25] act_dcollect_digit
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
vcsm_timer: 47173
vcsm_process_event: [1/0, 0.7, 25] act_dcollect_digit
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
vcsm_timer: 47197
vcsm_process_event: [1/0, 0.7, 25] act_dcollect_digit
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
vcsm_timer: 47217
vcsm_process_event: [1/0, 0.7, 13] act_dcollect_proc
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
dsp_digit_collect_off: [1/0] packet_len=10 channel_id=129 packet_id=36
dsp_idle_mode: [1/0] packet_len=10 channel_id=1 packet_id=68
```

The following output shows that the network voice path cuts through:

```
vcsm_process_event: [1/0, 0.8, 15] act_bridge
vcsm_process_event: [1/0, 0.8, 20] act_caps_ind
vcsm_process_event: [1/0, 0.8, 21] act_caps_ack
dsp_voice_mode: [1/0] packet_len=22 channel_id=1 packet_id=73 coding_type=6
voice_field_size=20 VAD_flag=1 echo_length=128 comfort_noise=1 fax_detect=1
```

The following output shows that the called-party end of the connection is connected:

```
vcsm_process_event: [1/0, 0.8, 8] act_connect
```

The following output shows the voice quality statistics collected periodically:

```
vcsm_process_event: [1/0, 0.13, 17]
dsp_get_rx_stats: [1/0] packet_len=12 channel_id=1 packet_id=87 reset_flag=0
vcsm_process_event: [1/0, 0.13, 28]
vcsm_process_event: [1/0, 0.13, 29]
vcsm_process_event: [1/0, 0.13, 32]
vcsm_process_event: [1/0, 0.13, 17]
dsp_get_rx_stats: [1/0] packet_len=12 channel_id=1 packet_id=87 reset_flag=0
vcsm_process_event: [1/0, 0.13, 28]
vcsm_process_event: [1/0, 0.13, 29]
vcsm_process_event: [1/0, 0.13, 32]
vcsm_process_event: [1/0, 0.13, 17]
dsp_get_rx_stats: [1/0] packet_len=12 channel_id=1 packet_id=87 reset_flag=0
vcsm_process_event: [1/0, 0.13, 28]
vcsm_process_event: [1/0, 0.13, 29]
vcsm_process_event: [1/0, 0.13, 32]
```

The following output shows that the disconnection indication is passed to higher level code. The call connection is torn down, and final call statistics are collected.

```
vcsm_process_event: [1/0, 0.13, 4] act_generate_disc
vcsm_process_event: [1/0, 0.13, 16] act_bdrop
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
vcsm_process_event: [1/0, 0.13, 18] act_disconnect
dsp_get_levels: [1/0] packet_len=10 channel_id=1 packet_id=89
vcsm_timer: 48762
vcsm_process_event: [1/0, 0.15, 34] act_get_levels
dsp_get_tx_stats: [1/0] packet_len=12 channel_id=1 packet_id=86 reset_flag=1
vcsm_process_event: [1/0, 0.15, 31] act_stats_complete
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
dsp_digit_collect_off: [1/0] packet_len=10 channel_id=129 packet_id=36
dsp_idle_mode: [1/0] packet_len=10 channel_id=1 packet_id=68
vcsm_timer: 48762
dsp_set_sig_state: [1/0] packet_len=14 channel_id=129 packet_id=39 state=0x4
timestamp=0x0
vcsm_process_event: [1/0, 0.16, 5] act_wrelease_release
dsp_CP_tone_off: [1/0] packet_len=10 channel_id=1 packet_id=71
dsp_idle_mode: [1/0] packet_len=10 channel_id=1 packet_id=68
dsp_get_rx_stats: [1/0] packet_len=12 channel_id=1 packet_id=87 reset_flag=1
```

# debug vtsp all

Use the **debug vtsp all** EXEC command to show debugging information for all of the **debug vtsp** commands. Use the **no** form of this command to disable debugging output.

**[no] debug vtsp all**

## Usage Guidelines

The **debug vtsp all** command enables the following debug voice telephony service provider (vtsp) commands: **debug vtsp session**, **debug vtsp error**, and **debug vtsp dsp**. For more information or sample output, refer to the individual commands in this chapter.

# debug vtsp dsp

Use the **debug vtsp dsp** EXEC command to show messages from the digital signal processor (DSP) on the V.Fast Class (VFC) modem to the router. Use the **no** form of this command to disable debugging output.

[**no**] **debug vtsp dsp**

### Usage Guidelines

The **debug vtsp dsp** command shows messages from the DSP on the VFC to the router; this command is useful if you suspect that the VFC is not functional. It is a simple way to check if the VFC is responding to off-hook indications.

### Sample Display

The following output shows the collection of DTMF digits from the DSP:

```
router# debug vtsp dsp
*Nov 30 00:44:34.491: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT: digit=3
*Nov 30 00:44:36.267: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT: digit=1
*Nov 30 00:44:36.571: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT: digit=0
*Nov 30 00:44:36.711: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT: digit=0
*Nov 30 00:44:37.147: vtsp_process_dsp_message: MSG_TX_DTMF_DIGIT: digit=2
```

# debug vtsp error

Use the **debug vtsp error** command to display processing errors in the voice telephony service provider. Use the **no** form of this command to disable vtsp error debugging.

[**no**] **debug vtsp error**

### Usage Guidelines

The **debug vtsp error** command can be used to check for mismatches in interface capabilities.

## Sample Display

The following example shows sample output from the **debug vtsp error** command, in which a dialed number is not reachable because it is not configured.

```
router# deb vtsp error
Voice telephony call control error debugging is on

router#
*Mar  1 00:21:48.698:cc_api_call_setup_ind (vdbPtr=0x1575AB0,
callInfo={called=,called_oct3=0x81,calling=9999,calling_oct3=0x0,called_oct3a=0x0,
       fdest=0 peer_tag=1},callID=0x15896A4)
*Mar  1 00:21:48.698:cc_api_call_setup_ind type 3 , prot 0
*Mar  1 00:21:48.706:cc_process_call_setup_ind (event=0x16AD0E0) handed call to app
"SESSION"
*Mar  1 00:21:48.706:sess_appl:ev(23=CC_EV_CALL_SETUP_IND), cid(15), disp(0)
*Mar  1 00:21:48.706:sess_appl:ev(SSA_EV_CALL_SETUP_IND), cid(15), disp(0)
*Mar  1 00:21:48.706:ccCallSetContext (callID=0xF, context=0x1632898)
*Mar  1 00:21:48.706:ccCallSetupAck (callID=0xF)
*Mar  1 00:21:48.706:ccGenerateTone (callID=0xF tone=8)
*Mar  1 00:21:49.710:cc_api_call_digit_begin (vdbPtr=0x1575AB0, callID=0xF, digit=5,
flags=0x1, timestamp=0xB1AE6BC4, expiration=0x0)
*Mar  1 00:21:49.710:sess_appl:ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(15), disp(0)
*Mar  1 00:21:49.710:cid(15)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(1)fDest(0)
*Mar  1 00:21:49.714:ssaIgnore cid(15), st(SSA_CS_MAPPING),oldst(0), ev(10)
*Mar  1 00:21:49.778:cc_api_call_digit (vdbPtr=0x1575AB0, callID=0xF, digit=5,
duration=4165,tag 0, callparty 0 )
*Mar  1 00:21:49.778:sess_appl:ev(9=CC_EV_CALL_DIGIT), cid(15), disp(0)
*Mar  1 00:21:49.778:cid(15)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(1)fDest(0)
*Mar  1 00:21:49.782:ssaDigit
*Mar  1 00:21:49.782:ssaDigit, callinfo , digit 5, tag 0,callparty 0
*Mar  1 00:21:49.782:ssaDigit, calling 9999,result 1
*Mar  1 00:21:49.915:cc_api_call_digit_begin (vdbPtr=0x1575AB0, callID=0xF, digit=5,
flags=0x1, timestamp=0xB1AF6B6C, expiration=0x0)
*Mar  1 00:21:49.915:sess_appl:ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(15), disp(0)
*Mar  1 00:21:49.915:cid(15)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(1)fDest(0)
*Mar  1 00:21:49.915:ssaIgnore cid(15), st(SSA_CS_MAPPING),oldst(0), ev(10)
*Mar  1 00:21:49.999:cc_api_call_digit (vdbPtr=0x1575AB0, callID=0xF, digit=5,
duration=95,tag 0, callparty 0 )
*Mar  1 00:21:49.999:sess_appl:ev(9=CC_EV_CALL_DIGIT), cid(15), disp(0)
*Mar  1 00:21:50.003:cid(15)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(1)fDest(0)
*Mar  1 00:21:50.003:ssaDigit
*Mar  1 00:21:50.003:ssaDigit, callinfo , digit 55, tag 0,callparty 0
*Mar  1 00:21:50.003:ssaDigit, calling 9999,result -1
*Mar  1 00:21:50.003:ccCallDisconnect (callID=0xF, cause=0x1C tag=0x0)
*Mar  1 00:21:50.003:ccCallDisconnect (callID=0xF, cause=0x1C tag=0x0)
*Mar  1 00:21:50.007:vtsp_process_event():prev_state = 0.4 ,
 state = S_WAIT_RELEASE_NC, event = E_CC_DISCONNECT
 Invalid FSM  Input on channel 1/1:15
*Mar  1 00:21:52.927:vtsp_process_event():prev_state = 0.7 ,
 state = S_WAIT_RELEASE_RESP, event = E_TSP_CALL_FEATURE_IND
 Invalid FSM  Input on channel 1/1:15
*Mar  1 00:21:52.931:cc_api_call_disconnect_done(vdbPtr=0x1575AB0, callID=0xF, disp=0,
tag=0x0)
*Mar  1 00:21:52.931:sess_appl:ev(13=CC_EV_CALL_DISCONNECT_DONE), cid(15), disp(0)
*Mar  1 00:21:52.931:cid(15)st(SSA_CS_DISCONNECTING)ev(SSA_EV_CALL_DISCONNECT_DONE)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(1)fDest(0)
```

# debug vtsp port

To observe the behavior of the VTSP state machine on a specific voice port, use the **debug vtsp port** command. Use the **no** form of the command to turn off the debug function.

**For Cisco 1700 series with analog voice ports:**

**debug vtsp port** *slot/port*

**no debug vtsp port** *slot/port*

## Sytnax Description

**For the Cisco 1700 series with analog voice ports:**

| | |
|---|---|
| *slot/port* | Debugs the analog voice port you specify with the *slot/port* designation.<br>*slot* is the physical slot in which the analog voice interface card (VIC) is installed. Valid entries are 0, 1, and 2.<br>*port* specifies an analog voice port number within the analog VIC in the slot. Valid entries are 0 and 1. |

## Usage Guidelines

Use the **debug vtsp port** command to limit the debug output to a particular voice port. The debug output can be quite voluminous for a single channel. Use this debug with any or all of the other debug modes.

Execution of **no debug vtsp all** will turn off all VTSP-level debugging. It is usually a good idea to turn off all debugging and then enter the debug commands you are interested in one by one. This will help to avoid confusion about which ports you are actually debugging.

## Sample Display

The following example shows sample output from the **debug vtsp port 0/1** and **debug vtsp all** commands:

```
router# debug vtsp port 0/1

21:59:14: vtsp_tsp_call_setup_ind (sdb=0x816CCA34, tdm_info=0x0,
tsp_info=0x816CC600, calling_number= calling_oct3 = 0x0, called_number=
called_oct3 = 0x81, oct3a=0x0): peer_tag=201
21:59:14: : ev.clg.clir is 0
 ev.clg.clid_transparent is 0
 ev.clg.null_orig_clg is 1
 ev.clg.calling_translated is false

21:59:14: vtsp_do_call_setup_ind
21:59:14: vtsp_allocate_cdb,cdb 0x81313820
21:59:14: vtsp_do_normal_call_setup_ind
21:59:14: vtsp_insert_cdb,cdb 0x81313820
21:59:14: vtsp_open_voice_and_set_params
21:59:14: vtsp_modem_proto_from_cdb: cap_modem_proto 1073741824
21:59:14: vtsp_modem_proto_from_cdb: cap_modem_proto 1073741824playout default

21:59:14: vtsp_report_digit_control: enable=1: digit reporting enabled
21:59:14: : vtsp_get_digit_timeouts
21:59:14: vtsp:[0/1:5505, S_SETUP_INDICATED, E_CC_SETUP_ACK]
21:59:14: act_setup_ind_ack act_setup_ind_ack(): vtsp_dsp_dtmf_mode()

21:59:14: vtsp_modem_proto_from_cdb: cap_modem_proto 0
21:59:14: vtsp_modem_proto_from_cdb: cap_modem_proto 0act_setup_ind_ack:
modem_mode = 0, fax_relay_on = 1
21:59:14: act_setup_ind_ack(): dsp_dtmf_mode()

21:59:14: vtsp_timer: 7915452
21:59:14: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_CC_GEN_TONE]
21:59:14: act_gen_tone
21:59:14: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_CC_GEN_TONE]
21:59:14: act_gen_tone
21:59:15: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:15: act_report_digit_begin
21:59:15: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT]
21:59:15: act_report_digit_end
21:59:15: vtsp_timer_stop: 7915584
21:59:15: vtsp_timer: 7915584
21:59:15: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:15: act_report_digit_begin
21:59:16: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT]
21:59:16: act_report_digit_end
21:59:16: vtsp_timer_stop: 7915604
21:59:16: vtsp_timer: 7915604
21:59:16: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:16: act_report_digit_begin
21:59:16: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_DSP_DTMF_DIGIT]
21:59:16: act_report_digit_end
21:59:16: vtsp_timer_stop: 7915624
21:59:16: vtsp_timer: 7915624
21:59:16: vtsp_report_digit_control: enable=0: digit reporting disabled
21:59:16: : vtsp_get_digit_timeouts
21:59:16: vtsp_save_dialpeer_tag: tag = 221
21:59:16: vtsp:[0/1:5505, S_DIGIT_COLLECT, E_CC_PROCEEDING]
21:59:16: act_dcollect_proc
21:59:16: vtsp_do_call_setup_req
21:59:16: digit_strip:1, pcn:221, poa:221
21:59:16: pcn:, poa:
```

```
21:59:16: Final pcn:, poa:, dial_string:
21:59:16: vtsp_get_dialpeer_tag: tag = 221
21:59:16: vtsp_get_dialpeer_tag: tag = 221
21:59:16: vtsp:[0/1:5505, S_PROCEEDING, E_CC_PROGRESS]
21:59:16: act_progress
21:59:16: vtsp_timer_stop: 7915625
21:59:16: vtsp:[0/1:5505, S_PROCEEDING, E_CC_BRIDGE]
21:59:16: act_bridge
21:59:16: vtsp_tdm_hpm_bridge
21:59:16: vtsp_tdm_hpm_bridge: cdb allow_tdm_hairpin = FALSE, dst_cdb_ptr
allow_tdm_hairpin = TRUE
21:59:16: vtsp:[0/1:5505, S_PROCEEDING, E_CC_CAPS_IND]
21:59:16: act_caps_ind playout default

21:59:16: act_caps_ind: passthrough: cap_modem_proto 1073741824, cap_modem_codec
0, cap_modem_redundancy 0, payload 79157256
21:59:16: act_caps_ind:Encap 1, Vad 2, Codec 0x1, CodecBytes 80,
            FaxRate 1, FaxBytes 20, FaxNsf 0x002A
            SignalType 2
            DtmfRelay 1, Modem 2, SeqNumStart 0x20B3
21:59:16: act_caps_ind: [ mode:0,init:60, min:4, max:200]
21:59:16: vtsp:[0/1:5505, S_PROCEEDING, E_CC_CAPS_ACK]
21:59:16: act_caps_ack
21:59:16: act_caps_ack: passthrough: cap_modem_proto 1073741824, cap_modem_codec
0, cap_modem_redundancy 0, payload 79157256
21:59:16: act_switch_codec: codec = 5

21:59:16: vtsp_modem_proto_from_cdb: cap_modem_proto 1073741824
21:59:16: vtsp_rtp_nse_payload_from_cdb: payload 100
21:59:16: vtsp_modem_proto_from_cdb: cap_modem_proto 1073741824
21:59:18: vtsp_get_dialpeer_tag: tag = 221
21:59:18: vtsp:[0/1:5505, S_PROCEEDING, E_CC_CONNECT]
21:59:18: act_connect
21:59:18: vtsp_ring_noan_timer_stop: 7915855
21:59:22: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:22: act_report_digit_begin
21:59:22: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT]
21:59:22: act_report_digit_end
21:59:22: vtsp_timer_stop: 7916256
21:59:22: vtsp_timer: 7916256
21:59:25: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:25: act_report_digit_begin
21:59:25: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT]
21:59:25: act_report_digit_end
21:59:25: vtsp_timer_stop: 7916576
21:59:25: vtsp_timer: 7916576
21:59:28: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:28: act_report_digit_begin
21:59:28: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT]
21:59:28: act_report_digit_end
21:59:28: vtsp_timer_stop: 7916896
21:59:28: vtsp_timer: 7916896
21:59:31: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:31: act_report_digit_begin
21:59:32: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT]
21:59:32: act_report_digit_end
21:59:32: vtsp_timer_stop: 7917216
21:59:32: vtsp_timer: 7917216
21:59:35: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:35: act_report_digit_begin
21:59:35: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT]
21:59:35: act_report_digit_end
21:59:35: vtsp_timer_stop: 7917536
21:59:35: vtsp_timer: 7917536
```

```
21:59:38: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT_BEGIN]
21:59:38: act_report_digit_begin
21:59:38: vtsp:[0/1:5505, S_CONNECT, E_DSP_DTMF_DIGIT]
21:59:38: act_report_digit_end
21:59:38: vtsp_timer_stop: 7917856
21:59:38: vtsp_timer: 7917856
21:59:39: vtsp:[0/1:5505, S_CONNECT, E_TSP_DISCONNECT_IND]
21:59:39: act_generate_disc
21:59:39: vtsp_ring_noan_timer_stop: 7917977
21:59:39: vtsp_timer_stop: 7917977
21:59:39: vtsp_pcm_tone_detect_timer_stop: 7917977
21:59:39: vtsp:[0/1:5505, S_CONNECT, E_CC_BRIDGE_DROP]
21:59:39: act_bdrop
21:59:39: vtsp:[0/1:5505, S_CONNECT, E_CC_DISCONNECT]
21:59:39: act_disconnect
21:59:39: vtsp_ring_noan_timer_stop: 7917977
21:59:39: vtsp_pcm_tone_detect_timer_stop: 7917977
21:59:39: vtsp_pcm_switchover_timer_stop: 7917977
21:59:39: vtsp_timer_stop: 7917977
21:59:39: vtsp_timer: 7917977
21:59:39: vtsp:[0/1:5505, S_WAIT_STATS, E_DSP_GET_ERROR]
21:59:39: act_get_error
21:59:39: vtsp_print_error_stats: rx_dropped=0 tx_dropped=0 rx_control=40
tx_control=20 tx_control_dropped=0  dsp_mode_channel_1=0 dsp_mode_channel_2=0
c[0]=76 c[1]=68 c[2]=68 c[3]=78 c[4]=106 c[5]=92 c[6]=73 c[7]=71 c[8]=71 c[9]=71
c[10]=71 c[11]=71 c[12]=71 c[13]=68 c[14]=73 c[15]=6
21:59:39: vtsp_timer_stop: 7917978
21:59:39: vtsp_timer: 7917978
21:59:39: vtsp:[0/1:5505, S_WAIT_STATS, E_DSP_GET_LEVELS]
21:59:39: act_get_levels
21:59:39: vtsp:[0/1:5505, S_WAIT_STATS, E_DSP_GET_TX]
21:59:39: act_stats_complete
21:59:39: vtsp_timer_stop: 7917978
21:59:39: vtsp_ring_noan_timer_stop: 7917978
21:59:39: vtsp_timer: 7917978
21:59:39: vtsp:[0/1:5505, S_WAIT_RELEASE, E_TSP_DISCONNECT_CONF]
21:59:39: act_wrelease_release
21:59:39: vtsp_timer_stop: 7917978vtsp_do_call_historyvtsp_do_call_history
CoderRate 5
21:59:39: vtsp:[0/1:5505, S_CLOSE_DSPRM, E_DSPRM_CLOSE_COMPLETE]
21:59:39: act_terminate
```

# debug vtsp session

Use the **debug vtsp session** EXEC command to trace how the router interacts with the digital signal processor (DSP) based on the signaling indications from the signaling stack and requests from the application. Use the **no** form of this command to disable debugging output.

[**no**] **debug vtsp session**

## Usage Guidelines

The **debug vtsp session** command displays information about how each network indication and application request is processed, signaling indications, and DSP control messages.

This debug level shows the internal workings of the voice telephony call state machine.

### Sample Display

The following output shows that the call has been accepted and that the system is now checking for incoming dial-peer matches:

```
router# debug vtsp session
*Nov 30 00:46:19.535: vtsp_tsp_call_accept_check (sdb=0x60CD4C58,
calling_number=408 called_number=1): peer_tag=0
*Nov 30 00:46:19.535: vtsp_tsp_call_setup_ind (sdb=0x60CD4C58,
tdm_info=0x60B80044, tsp_info=0x60B09EB0, calling_number=408 called_number=1):
peer_tag=1
```

The following output shows that a DSP has been allocated to process the call and indicate to the higher layer code:

```
*Nov 30 00:46:19.535: vtsp_do_call_setup_ind:
*Nov 30 00:46:19.535: dsp_open_voice_channel: [0:D:12] packet_len=12
channel_id=8737 packet_id=74 alaw_ulaw_select=0 transport_protocol=2
*Nov 30 00:46:19.535: dsp_set_playout_delay: [0:D:12] packet_len=18
channel_id=8737 packet_id=76 mode=1 initial=60 min=4 max=200 fax_nom=300
*Nov 30 00:46:19.535: dsp_echo_canceller_control: [0:D:12] packet_len=10
channel_id=8737 packet_id=66 flags=0x0
*Nov 30 00:46:19.539: dsp_set_gains: [0:D:12] packet_len=12 channel_id=8737
packet_id=91 in_gain=0 out_gain=0
*Nov 30 00:46:19.539: dsp_vad_enable: [0:D:12] packet_len=10 channel_id=8737
packet_id=78 thresh=-38
*Nov 30 00:46:19.559: vtsp_process_event: [0:D:12, 0.3, 13] act_setup_ind_ack
```

The following output shows that the higher layer code has accepted the call, placed the DSP in dual tone multifrequency (DTMF) mode, and collected digits:

```
*Nov 30 00:46:19.559: dsp_voice_mode: [0:D:12] packet_len=20 channel_id=8737
packet_id=73 coding_type=1 voice_field_size=160 VAD_flag=0 echo_length=64
comfort_noise=1 fax_detect=1
*Nov 30 00:46:19.559: dsp_dtmf_mode: [0:D:12] packet_len=10 channel_id=8737
packet_id=65 dtmf_or_mf=0
*Nov 30 00:46:19.559: dsp_cp_tone_on: [0:D:12] packet_len=30 channel_id=8737
packet_id=72 tone_id=3 n_freq=2 freq_of_first=350 freq_of_second=440
amp_of_first=4000 amp_of_second=4000 direction=1 on_time_first=65535
off_time_first=0 on_time_second=65535 off_time_second=0
*Nov 30 00:46:19.559: vtsp_timer: 278792
*Nov 30 00:46:22.059: vtsp_process_event: [0:D:12, 0.4, 25] act_dcollect_digit
*Nov 30 00:46:22.059: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:22.059: vtsp_timer: 279042
*Nov 30 00:46:22.363: vtsp_process_event: [0:D:12, 0.4, 25] act_dcollect_digit
*Nov 30 00:46:22.363: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:22.363: vtsp_timer: 279072
*Nov 30 00:46:22.639: vtsp_process_event: [0:D:12, 0.4, 25] act_dcollect_digit
*Nov 30 00:46:22.639: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:22.639: vtsp_timer: 279100
*Nov 30 00:46:22.843: vtsp_process_event: [0:D:12, 0.4, 25] act_dcollect_digit
*Nov 30 00:46:22.843: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:22.843: vtsp_timer: 279120
*Nov 30 00:46:23.663: vtsp_process_event: [0:D:12, 0.4, 25] act_dcollect_digit
*Nov 30 00:46:23.663: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:23.663: vtsp_timer: 279202
```

The following output shows that the call proceeded and that DTMF was disabled:

```
*Nov 30 00:46:23.663: vtsp_process_event: [0:D:12, 0.4, 15] act_dcollect_proc
*Nov 30 00:46:23.663: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:23.663: dsp_idle_mode: [0:D:12] packet_len=8 channel_id=8737
packet_id=68
```

The following output shows that the telephony call leg was conferenced with the packet network call leg and performed capabilities exchange with the network-side call leg:

```
*Nov 30 00:46:23.699: vtsp_process_event: [0:D:12, 0.5, 17] act_bridge
*Nov 30 00:46:23.699: vtsp_process_event: [0:D:12, 0.5, 22] act_caps_ind
*Nov 30 00:46:23.699: vtsp_process_event: [0:D:12, 0.5, 23] act_caps_ack
Go into voice mode with codec indicated in caps exchange.
*Nov 30 00:46:23.699: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:23.699: dsp_idle_mode: [0:D:12] packet_len=8 channel_id=8737
packet_id=68
*Nov 30 00:46:23.699: dsp_voice_mode: [0:D:12] packet_len=20 channel_id=8737
packet_id=73 coding_type=6 voice_field_size=20 VAD_flag=1 echo_length=64
comfort_noise=1 fax_detect=1
```

The following output shows the call connected at remote side:

```
*Nov 30 00:46:23.779: vtsp_process_event: [0:D:12, 0.5, 10] act_connect
```

The following output shows that disconnect was indicated, and passed to upper layers:

```
*Nov 30 00:46:30.267: vtsp_process_event: [0:D:12, 0.11, 5] act_generate_disc
```

The following output shows that the conference was torn down and disconnect handshake completed:

```
*Nov 30 00:46:30.267: vtsp_process_event: [0:D:12, 0.11, 18] act_bdrop
*Nov 30 00:46:30.267: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:30.267: vtsp_process_event: [0:D:12, 0.11, 20] act_disconnect
*Nov 30 00:46:30.267: dsp_get_error_stat: [0:D:12] packet_len=10 channel_id=0
packet_id=6 reset_flag=1
*Nov 30 00:46:30.267: vtsp_timer: 279862
```

The following output shows that the final DSP statistics were retrieved:

```
*Nov 30 00:46:30.275: vtsp_process_event: [0:D:12, 0.17, 30] act_get_error
*Nov 30 00:46:30.275: 0:D:12: rx_dropped=0 tx_dropped=0 rx_control=353
tx_control=338 tx_control_dropped=0 dsp_mode_channel_1=2 dsp_mode_channel_2=0
c[0]=71 c[1]=71 c[2]=71 c[3]=71 c[4]=68 c[5]=71 c[6]=68 c[7]=73 c[8]=83 c[9]=84
c[10]=87 c[11]=83 c[12]=84 c[13]=87 c[14]=71 c[15]=6
*Nov 30 00:46:30.275: dsp_get_levels: [0:D:12] packet_len=8 channel_id=8737
packet_id=89
*Nov 30 00:46:30.279: vtsp_process_event: [0:D:12, 0.17, 34] act_get_levels
*Nov 30 00:46:30.279: dsp_get_tx_stats: [0:D:12] packet_len=10 channel_id=8737
packet_id=86 reset_flag=1
*Nov 30 00:46:30.287: vtsp_process_event: [0:D:12, 0.17, 31] act_stats_complete
*Nov 30 00:46:30.287: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:30.287: dsp_idle_mode: [0:D:12] packet_len=8 channel_id=8737
packet_id=68
*Nov 30 00:46:30.287: vtsp_timer: 279864
```

The following output shows that the DSP channel was closed and released:

```
*Nov 30 00:46:30.287: vtsp_process_event: [0:D:12, 0.18, 6] act_wrelease_release
*Nov 30 00:46:30.287: dsp_cp_tone_off: [0:D:12] packet_len=8 channel_id=8737
packet_id=71
*Nov 30 00:46:30.287: dsp_idle_mode: [0:D:12] packet_len=8 channel_id=8737
packet_id=68
*Nov 30 00:46:30.287: dsp_close_voice_channel: [0:D:12] packet_len=8
channel_id=8737 packet_id=75
*Nov 30 00:46:30.287: vtsp_process_event: [0:D:12, 0.16, 42] act_terminate
```

# debug vtsp stats

Use the **debug vtsp stats** EXEC command to debug periodic messages sent and received from the **digital signal processor (**DSP) requesting statistical information during the call. Use the **no** form of this command to disable debugging output.

[**no**] **debug vtsp stats**

## Usage Guidelines

The **debug vtsp stats** command generates a collection of DSP statistics for generating RTP Control Protocol (RTCP) packets and a collection of other statistical information.

## Sample Display

The following output shows sample **debug vtsp stats** output:

```
router# debug vtsp stats
*Nov 30 00:53:26.499: vtsp_process_event: [0:D:14, 0.11, 19] act_packet_stats
*Nov 30 00:53:26.499: dsp_get_voice_playout_delay_stats: [0:D:14] packet_len=10
channel_id=8753 packet_id=83 reset_flag=0
*Nov 30 00:53:26.499: dsp_get_voice_playout_error_stats: [0:D:14] packet_len=10
channel_id=8753 packet_id=84 reset_flag=0
*Nov 30 00:53:26.499: dsp_get_rx_stats: [0:D:14] packet_len=10 channel_id=8753
packet_id=87 reset_flag=0
*Nov 30 00:53:26.503: vtsp_process_dsp_message: MSG_TX_GET_VOICE_PLAYOUT_DELAY:
clock_offset=-1664482334 curr_rx_delay_estimate=69 low_water_mark_rx_delay=69
high_water_mark_rx_delay=70
*Nov 30 00:53:26.503: vtsp_process_event: [0:D:14, 0.11, 28]
act_packet_stats_res
*Nov 30 00:53:26.503: vtsp_process_dsp_message: MSG_TX_GET_VOICE_PLAYOUT_ERROR:
predective_concelement_duration=0 interpolative_concelement_duration=0
silence_concelement_duration=0 retroactive_mem_update=0
buf_overflow_discard_duration=10 num_talkspurt_detection_errors=0
*Nov 30 00:53:26.503: vtsp_process_event: [0:D:14, 0.11, 29]
act_packet_stats_res
*Nov 30 00:53:26.503: vtsp_process_dsp_message: MSG_TX_GET_RX_STAT:
num_rx_pkts=152 num_early_pkts=-2074277660 num_late_pkts=327892
num_signaling_pkts=0 num_comfort_noise_pkts=0 receive_durtation=3130
voice_receive_duration=2970 fax_receive_duration=0 num_pack_ooseq=0
num_bad_header=0
*Nov 30 00:53:26.503: vtsp_process_event: [0:D:14, 0.11, 32]
act_packet_stats_res
```

# debug vtsp tone

To display debug messages showing the types of tones generated by the VoIP gateway, use the **debug vtsp tone** command. To disable the debug messages, use the **no** form of this command.

**[no] debug vtsp tone**

### Sample Display

The following example shows that a ringback tone was generated by the VoIP gateway:

```
Router# debug vtsp tone
*Jan  1 16:33:52.395:act_alert:Tone Ring Back generated in direction Network

*Jan  1 16:33:52.399:ISDN Se0:23:TX ->  ALERTING pd = 8  callref = 0x9816
```

# debug vtsp vofr subframe

To display the first 10 bytes (including header) of selected VoFR subframes for the interface, use the **debug vtsp vofr subframe** command. Use the **no** form of the command to turn off the debug function.

[**no**] **debug vtsp vofr subframe** *payload* [**from-dsp**] [**to-dsp**]

### Syntax Description

| | |
|---|---|
| *payload* | Number used to selectively display subframes of a specific payload. Payload types are:<br>0: Primary Payload - WARNING! This option might cause network instability<br>**1**: Annex-A<br>**2**: Annex-B<br>**3**: Annex-D<br>**4**: All other payloads<br>**5**: All payloads - WARNING! This option may cause network instability |
| **from-dsp** | Displays only the subframes received from the DSP. |
| **to-dsp** | Displays only the subframes going to the DSP. |

### Usage Guidelines

Each debug output displays the first 10 bytes of the FRF.11 subframe, including header bytes. The **from-dsp** and **to-dsp** options can be used to limit the debugs to a single direction. If not specified, debugs are displayed for subframes when they are received from the DSP and before they are sent to the DSP.

Use extreme caution in selecting payload options 0 and 6. These options may cause network instability.

### Sample Display

The following example shows sample output from the **debug vtsp vofr subframe** command:

```
router# debug vtsp vofr subframe 2
vtsp VoFR subframe debugging is enabled for payload 2 to and from DSP 3620_vofr#
*Mar  6 18:21:17.413:VoFR frame received from Network (24 bytes):9E 02 19 AA AA AA AA
      AA AA AA
*Mar  6 18:21:17.449:VoFR frame received from DSP (18 bytes):9E 02 19 AA AA AA AA AA AA
      AA
*Mar  6 18:21:23.969:VoFR frame received from Network (24 bytes):9E 02 19 AA AA AA AA
      AA AA AA
*Mar  6 18:21:24.005:VoFR frame received from DSP (18 bytes):9E 02 19 AA AA AA AA AA AA
      AA
```

# 6

# Routing Between Virtual LANs Overview

This chapter provides an overview of virtual LANs (VLANs). It describes the encapsulation protocols used for routing between VLANs and provides some basic information about designing VLANs.

This chapter describes VLANs. It contains the following sections:

- What Is a VLAN?
- VLAN Colors
- Why Implement VLANs?
- Communicating Between VLANs
- Designing Switched VLANs

## What Is a VLAN?

A VLAN is a switched network that is logically segmented on an organizational basis, by functions, project teams, or applications rather than on a physical or geographical basis. For example, all workstations and servers used by a particular workgroup team can be connected to the same VLAN, regardless of their physical connections to the network or the fact that they might be intermingled with other teams. Reconfiguration of the network can be done through software rather than by physically unplugging and moving devices or wires.

A VLAN can be thought of as a broadcast domain that exists within a defined set of switches. A VLAN consists of a number of end systems, either hosts or network equipment (such as bridges and routers), connected by a single bridging domain. The bridging domain is supported on various pieces of network equipment; for example, LAN switches that operate bridging protocols between them with a separate bridge group for each VLAN.

VLANs are created to provide the segmentation services traditionally provided by routers in LAN configurations. VLANs address scalability, security, and network management. Routers in VLAN topologies provide broadcast filtering, security, address summarization, and traffic flow management. None of the switches within the defined group will bridge any frames, not even broadcast frames, between two VLANs. Several key issues need to be considered when designing and building switched LAN internetworks.

- LAN Segmentation

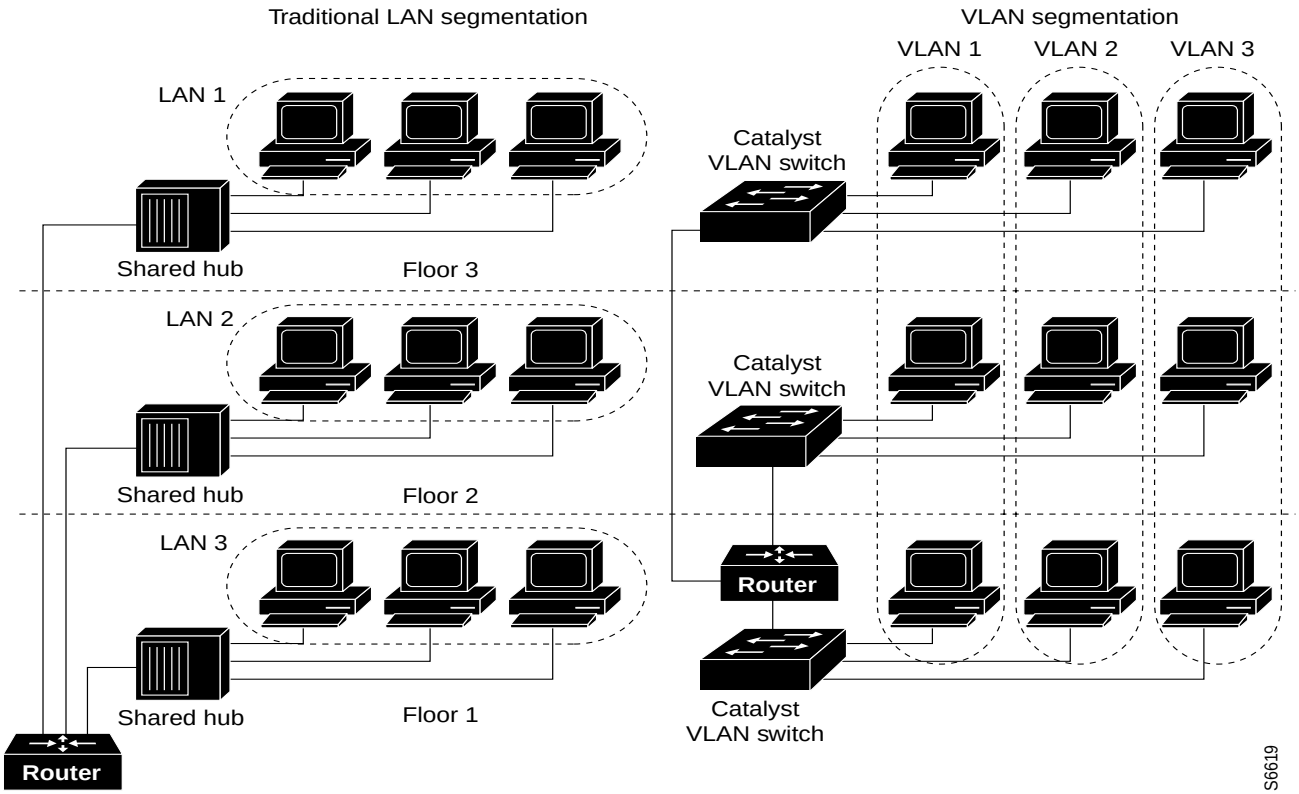
- Security
- Broadcast Control
- Performance
- Network Management
- Communication Between VLANs

# LAN Segmentation

VLANs allow logical network topologies to overlay the physical switched infrastructure such that any arbitrary collection of LAN ports can be combined into an autonomous user group or community of interest. The technology logically segments the network into separate Layer 2 broadcast domains whereby packets are switched between ports designated to be within the same VLAN. By containing traffic originating on a particular LAN only to other LANs in the same VLAN, switched virtual networks avoid wasting bandwidth, a drawback inherent to traditional bridged and switched networks in which packets are often forwarded to LANs with no need for them. Implementation of VLANs also improves scalability, particularly in LAN environments that support broadcast- or multicast-intensive protocols and applications that flood packets throughout the network.

illustrates the difference between traditional physical LAN segmentation and logical VLAN segmentation.

***Table 1 LAN Segmentation and VLAN Segmentation***



## Security

VLANs also improve security by isolating groups. High-security users can be grouped into a VLAN, possible on the same physical segment, and no users outside that VLAN can communicate with them.

## Broadcast Control

Just as switches isolate collision domains for attached hosts and only forward appropriate traffic out a particular port, VLANs provide complete isolation between VLANs. A VLAN is a bridging domain and all broadcast and multicast traffic is contained within it.

## Performance

The logical grouping of users allows an accounting group to make intensive use of a networked accounting system assigned to a VLAN that contains just that accounting group and its servers. That group's work will not affect other users. The VLAN configuration improves general network performance by not slowing down other users sharing the network.

## Network Management

The logical grouping of users allows easier network management. It is not necessary to pull cables to move a user from one network to another. Adds, moves, and changes are achieved by configuring a port into the appropriate VLAN.

## Communication Between VLANs

Communication between VLANs is accomplished through routing, and the traditional security and filtering functions of the router can be used. Cisco IOS software provides network services such as security filtering, quality of service (QoS), and accounting on a per VLAN basis. As switched networks evolve to distributed VLANs, Cisco IOS provides key inter-VLAN communications and allows the network to scale.

# VLAN Colors

VLAN switching is accomplished through *frame tagging* where traffic originating and contained within a particular virtual topology carries a unique VLAN identifier (VLAN ID) as it traverses a common backbone or trunk link. The VLAN ID enables VLAN switching devices to make intelligent forwarding decisions based on the embedded VLAN ID. Each VLAN is differentiated by a *color*, or VLAN identifier. The unique VLAN ID determines the *frame coloring* for the VLAN. Packets originating and contained within a particular VLAN carry the identifier that uniquely defines that VLAN (by the VLAN ID).

The VLAN ID allows VLAN switches and routers to selectively forward packets to ports with the same VLAN ID. The switch that receives the frame from the source station inserts the VLAN ID and the packet is switched onto the shared backbone network. When the frame exits the switched LAN, a switch

strips header and forwards the frame to interfaces that match the VLAN color. If you are using a Cisco network management product such as VlanDirector, you can actually color code the VLANs and monitor VLAN graphically.

# Why Implement VLANs?

Network managers can group logically networks that span all major topologies, including high-speed technologies such as, ATM, FDDI, and Fast Ethernet. By creating virtual LANs, system and network administrators can control traffic patterns and react quickly to relocations and keep up with constant changes in the network due to moving requirements and node relocation just by changing the VLAN member list in the router configuration. They can add, remove, or move devices or make other changes to network configuration using software to make the changes.

Benefits and drawbacks of creating VLANs should be considered when you design your network, including these issues:

- Scalability
- Performance improvements
- Security
- Network additions, moves, and changes

# Communicating Between VLANs

The Cisco 1751 router uses the IEEE 802.1Q protocol for routing between VLANs.

The IEEE 802.1Q protocol is used to interconnect multiple switches and routers and for defining VLAN topologies. IEEE 802.1Q support is currently available only for Fast Ethernet interfaces.

Procedures for configuring routing between VLANs with IEEE 802.1Q encapsulation are provided in the “Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation” chapter later in this publication.

## VLAN Translation

VLAN translation refers to the ability of the Cisco IOS software to translate between different virtual LANs or between VLAN and non-VLAN encapsulating interfaces at Layer 2. Translation is typically used for selective inter-VLAN switching of non-routable protocols and to extend a single VLAN topology across hybrid switching environments. It is also possible to bridge VLANs on the main interface; the VLAN encapsulating header is preserved. Topology changes in one VLAN domain do not affect a different VLAN.

# Designing Switched VLANs

By the time you are ready to configure routing between VLANs, you will have already defined them through the switches in your network. Issues related to network design and VLAN definition should be addressed during your network design. Refer to the *Cisco Internetworking Design Guide* and appropriate switch documentation for information on these topics:

- Sharing resources between VLANs
- Load Balancing
- Redundant Links
- Addressing
- Segmenting Networks with VLANs

  Segmenting the network into broadcast groups improves network security. Use router access lists based on station addresses, application types, and protocol types.

- Routers and their Role in Switched Networks

  In switched networks, routers perform broadcast management, route processing and distribution, and provide communications between VLANs. Routers provide VLAN access to shared resources and connect to other parts of the network that are either logically segmented with the more traditional subnet approach or that require access to remote sites across wide-area links.

7

# Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation

This chapter describes the required and optional tasks for configuring routing between VLANs with IEEE 802.1Q encapsulation. For a complete description of VLAN commands used in this chapter, refer to the "Cisco IOS Switching Commands" chapter in the *Cisco IOS Switching Services Command Reference*. For documentation of other commands that appear in this chapter, you can use the command reference master index or search online.

The IEEE 802.1Q protocol is used to interconnect multiple switches and routers and for defining VLAN topologies. IEEE 802.1Q support is currently available for Fast Ethernet interfaces.

## IEEE 802.1Q Encapsulation Configuration Task List

You can configure routing between any number of VLANs in your network. This section documents the configuration tasks for each protocol supported with IEEE 802.1Q encapsulation. The basic process is the same, regardless of the protocol being routed. It involves:

- Enabling the protocol on the router.
- Enabling the protocol on the interface.
- Defining the encapsulation format as IEEE 802.1Q.
- Customizing the protocol according to the requirements for your environment.

The configuration processes documented in this chapter include the following:

- Configuring AppleTalk Routing over IEEE 802.1Q
- Configuring IP Routing over IEEE 802.1Q
- Configuring IPX Routing over IEEE 802.1Q

### Configuring AppleTalk Routing over IEEE 802.1Q

AppleTalk can be routed over virtual LAN (VLAN) subinterfaces using the IEEE 802.1Q VLAN encapsulation protocol. AppleTalk Routing provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

To route AppleTalk over IEEE 802.1Q between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform these tasks in the order in which they appear:

- Enabling AppleTalk Routing
- Defining the VLAN Encapsulation Format
- Configuring AppleTalk on the Subinterface

## Enabling AppleTalk Routing

To enable AppleTalk routing on IEEE 802.1Q interfaces, use the following command in global configuration mode:

| Command | Purpose |
|---|---|
| **appletalk routing** [**eigrp** *router-number*] | Enables AppleTalk routing globally. |

**Note** For more information on configuring AppleTalk, see the "Configuring AppleTalk" chapter in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

## Configuring AppleTalk on the Subinterface

After you enable AppleTalk globally and define the encapsulation format, you need to enable it on the subinterface by specifying the cable range and naming the AppleTalk zone for each interface. To enable the AppleTalk protocol on the subinterface, use the following commands in interface configuration mode:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **appletalk cable-range** *cable-range* [*network.node*] | Assigns the AppleTalk cable range and zone for the subinterface. |
| **Step 2** | **appletalk zone** *zone-name* | Assigns the AppleTalk zone for the subinterface. |

## Defining the VLAN Encapsulation Format

To define the VLAN encapsulation format as IEEE 802.1Q, use the following commands in interface configuration mode:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **interface fastethernet** *slot*/*port.subinterface-number* | Specifies the subinterface the VLAN will use. |
| **Step 2** | **encapsulation dot1q** *vlan-identifier* | Defines the encapsulation format as IEEE 802.1Q (**dot1q**), and specify the VLAN identifier. |

# Configuring IP Routing over IEEE 802.1Q

IP routing over IEEE 802.1Q extends IP routing capabilities to include support for routing IP frame types in VLAN configurations using the IEEE 802.1Q encapsulation.

To route IP over IEEE 802.1Q between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform these tasks in the order in which they appear:

- Enabling IP Routing
- Defining the VLAN Encapsulation Format
- Assigning IP Address to Network Interface

## Enabling IP Routing

IP routing is automatically enabled in the Cisco IOS software for routers. To reenable IP routing if it has been disabled, use the following command in global configuration mode:

| Command | Purpose |
|---|---|
| **ip routing** | Enables IP routing on the router. |

Once you have IP routing enabled on the router, you can customize the characteristics to suit your environment. If necessary, refer to the IP configuration chapters in the *Cisco IOS IP and IP Routing Configuration Guide* for guidelines on configuring IP.

## Defining the VLAN Encapsulation Format

To define the encapsulation format as IEEE 802.1Q, use the following commands in interface configuration mode:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **interface fastethernet** *slot/port.subinterface-number* | Specifies the subinterface on which IEEE 802.1Q will be used. |
| **Step 2** | **encapsulation dot1q** *vlanid* | Defines the encapsulation format as IEEE 802.1Q (**dot1q**), and specify the VLAN identifier |

## Assigning IP Address to Network Interface

An interface can have one primary IP address. To assign a primary IP address and a network mask to a network interface, use the following command in interface configuration mode:

| Command | Purpose |
|---|---|
| **ip address** *ip-address mask* | Sets a primary IP address for an interface. |

A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a *subnet mask*.

# Configuring IPX Routing over IEEE 802.1Q

IPX Routing over IEEE 802.1Q VLANs extends Novell NetWare routing capabilities to include support for routing Novell Ethernet_802.3 encapsulation frame types in VLAN configurations. Users with Novell NetWare environments can configure Novell Ethernet_802.3 encapsulation frames to be routed using IEEE 802.1Q encapsulation across VLAN boundaries.

To configure Cisco IOS software on a router with connected VLANs to exchange IPX Novell Ethernet_802.3 encapsulated frames, perform these tasks in the order in which they are appear:

- Enabling NetWare Routing
- Defining the VLAN Encapsulation Format
- Configuring NetWare on the Subinterface

## Enabling NetWare Routing

To enable IPX routing on IEEE 802.1Q interfaces, use the following command in global configuration mode:

| Command | Purpose |
|---|---|
| **ipx routing** [*node*] | Enables IPX routing globally. |

## Defining the VLAN Encapsulation Format

To define the encapsulation format as IEEE 802.1Q, use the following commands in interface configuration mode:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **interface fastethernet** *slot/port.subinterface-number* | Specifies the subinterface on which IEEE 802.1Q will be used. |
| **Step 2** | **encapsulation dot1q** *vlan-identifier* | Defines the encapsulation format as IEEE 802.1Q and specify the VLAN identifier. |

## Configuring NetWare on the Subinterface

After you enable NetWare globally and define the VLAN encapsulation format, you may need to enable the subinterface by specifying the NetWare network number. Use this command in interface configuration mode:

| Command | Purpose |
|---|---|
| **ipx network** *network* | Specifies the IPX network number. |

# IEEE 802.1Q Encapsulation Configuration Examples

This section provides configuration examples for each of the protocols described in this feature guide. It includes these examples:

- Configuring AppleTalk over IEEE 802.1Q Example
- Configuring IP Routing over IEEE 802.1Q Example
- Configuring IPX Routing over IEEE 802.1Q Example

## Configuring AppleTalk over IEEE 802.1Q Example

This configuration example shows AppleTalk being routed on VLAN 100.

```
!
appletalk routing
!
interface fastethernet 0/0.100
  encapsulation dot1q 100
  appletalk cable-range 100-100 100.1
  appletalk zone eng
!
```

## Configuring IP Routing over IEEE 802.1Q Example

This configuration example shows IP being routed on VLAN 101.

```
!
ip routing
!
interface fastethernet 0/0.101
  encapsulation dot1q 101
  ip addr 10.0.0.11 255.0.0.0
!
```

## Configuring IPX Routing over IEEE 802.1Q Example

This configuration example shows IPX being routed on VLAN 102.

```
!
ipx routing
!
interface fastethernet 0/0.102
  encapsulation dot1q 102
  ipx network 100
!
```

# VLAN Commands

This section provides an alphabetical listing of all the VLAN commands that are new or specific to the Cisco 1751 router. All other commands used with this feature are documented in the Cisco IOS Release 12.1T command reference documents.

## clear vlan statistics

To remove virtual LAN statistics from any statically or system configured entries, use the **clear vlan statistics** privileged EXEC command.

**clear vlan statistics**

### Syntax Description

This command has no arguments or keywords.

### Default

No default behavior or values.

### Command Mode

Privileged EXEC

### Example

The following example clears VLAN statistics:

```
clear vlan statistics
```

## debug vlan packet

Use the **debug vlan packet** privileged EXEC command to display general information on virtual LAN (VLAN) packets that the router received but is not configured to support. The **no** form of this command disables debugging output.

**debug vlan packet**
**no debug vlan packet**

### Syntax Description

This command has no arguments or keywords.

### Usage Guidelines

The **debug vlan packet** command displays only packets with a VLAN identifier that the router is not configured to support. This command allows you to identify other VLAN traffic on the network. Virtual LAN packets that the router is configured to route or switch are counted and indicated when you use the **show vlans** command.

### Example

The following is sample output from the **debug vlan packet** output.

```
Router# debug vlan packet

Virtual LAN packet information debugging is on
```

## encapsulation dot1q

To enable IEEE 802.1Q encapsulation of traffic on a specified subinterface in virtual LANs, use the **encapsulation dot1q** command in subinterface configuration mode. IEEE 802.1Q is a standard protocol for interconnecting multiple switches and routers and for defining VLAN topologies.

**encapsulation dot1q** *vlan-id*

### Syntax Description

| | |
|---|---|
| *vlan-id* | Virtual LAN identifier. The allowed range is from 1 to 1000. |

### Default

Disabled

### Command Mode

Subinterface configuration

### Usage Guidelines

IEEE 802.1Q encapsulation is configurable on Fast Ethernet interfaces.

### Example

The following example encapsulates VLAN traffic using the IEEE 802.1Q protocol for VLAN 100:

```
interface fastethernet 0/0.100
  encapsulation dot1q 100
```

## show vlans

To view virtual LAN (VLAN) subinterfaces, use the **show vlans** privileged EXEC command.

**show vlans**

### Syntax Description

This command has no arguments or keywords.

### Command Mode

Privileged EXEC

### Example

The following is sample output from the **show vlans** command:

```
1751_2# show vlans

Virtual LAN ID:1 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface: FastEthernet0/0

This is configured as native Vlan for the following interface(s):
FastEthernet0/0

   Protocols Configured:   Address:           Received:        Transmitted:

Virtual LAN ID:100 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface: FastEthernet0/0.100

   Protocols Configured:   Address:           Received:        Transmitted:
      IP                   100.0.0.2               10                  10

Virtual LAN ID:2500 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface: FastEthernet0/0.200

   Protocols Configured:   Address:           Received:        Transmitted:
      IP                   200.0.0.2                5                   5
```

Table 1 describes the fields shown in the display.

***Table 1 show vlans Field Descriptions***

| Field | Description |
|---|---|
| Virtual LAN ID | Domain number of the VLAN. |
| vLAN Trunk Interface | Subinterface that carries the VLAN traffic. |
| Protocols Configured | Protocols configured on the VLAN. |
| Address | Network address. |
| Received | Packets received. |
| Transmitted | Packets transmitted. |

# GLOSSARY

## A

**ACOM** — Term used in G.165, "General Characteristics of International Telephone Connections and International Telephone Circuits: Echo Cancellers." ACOM is the combined loss achieved by the echo canceller, which is the sum of the Echo Return Loss, Echo Return Loss Enhancement, and nonlinear processing loss for the call.

**ADPCM** — Adaptive differential pulse code modulation. Process by which analog voice samples are encoded into high-quality digital signals.

**API** — Application programming interface. Specification of function-call conventions that defines an interface to a service.

## B

**BECN** — Backward explicit congestion notification. Bit set by a Frame Relay network in frames travelling in the opposite direction of frames encountering a congested path.

## C

**Call leg** — Segment of a call path. A logical connection between a telephone and a router, a router and a network, a router and a PBX, or a router and the PSTN using a session protocol. Each call leg corresponds to a dial peer.

**CIR** — Committed information rate. The average rate of information transfer a subscriber (for example, the network administrator) has stipulated for a Frame Relay PVC.

**CODEC** — Coder-decoder. Device that typically uses pulse code modulation to transform analog signals into a digital bit stream, and digital signals back into analog. In VoIP, it specifies the voice coder rate of speech for a dial peer.

## D

**Dial peer** — Software object that ties together a voice port and a local telephone number (*local dial peer* or *POTS dial peer*) or an IP address and a remote telephone number (*remote dial peer* or *VoIP dial peer*). Each dial peer corresponds to a call leg.

**DLCI** — Data-link connection identifier. Value that specifies a PVC or SVC in a Frame Relay network.

## D

**DSP** Digital signal processor. DSP segments the voice signal into frames and stores in voice packets.

**DTMF** Dual tone multifrequency. Use of two simultaneous voice-band tones for dialing (such as touch tone).

## E

**E.164** International public telecommunications numbering plan. A standard set by ITU-T that addresses telephone numbers.

**E&M** E&M interface uses a RJ-48 telephone cable to connect remote calls from an IP network to PBX trunk lines (tie lines) for local distribution. It is a signaling technique for two-wire and four-wire telephone and trunk interfaces.

## F

**Frame Relay** Industry standard for switched data link layer protocol that handles multiple virtual circuits using HDLC encapsulation between connected devices.

**FXO** Foreign exchange office. The FXO interface uses a RJ-11 modular telephone cable to connect local calls to a PSTN central office or to PBX that does not support E&M signaling. This interface is used for off-premise extension applications.

**FXS** Foreign exchange station. The FXS interface uses a standard RJ-11 modular telephone cable to connect directly to a standard telephone, fax machine, PBXs, or similar device, and supplies ring, voltage, and dial tone to the station.

## H

**H.323** ITU-T standard that describes packet-based video, audio, and data conferencing.

**HDLC** High-Level Data Link Control. A data link layer protocol that specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

## I

**ITU-T** International Telecommunications Union-Telecommunications standardization section.

## M

**Multilink PPP** Multilink Point-to-Point Protocol. This protocol defines a method of splitting, recombining, and sequencing datagrams across multiple logical data links.

## N

**NANP** North American Numbering Plan. The format in North America is 1Nxx-Nxx-xxxx, with N = digits 2 through 9 and x = digits 0 through 9.

## P

**PBX** Private branch exchange. Privately-owned central switching office.

**PCM** Pulse code modulation. Transmission of analog information in digital form through sampling and encoding the samples with a fixed number of bits.

**PLAR** Private line auto ringdown. PLAR connection associates a peer directly with an interface. This type of service results in a call attempt to some particular remote endpoint when the local extension is taken off-key.

**POTS** Plain old telephone service. Basic telephone service supplying standard single-line telephones, telephone lines, and access to the public switched telephone network.

**POTS dial peer** Dial peer connected via a traditional telephony network. A software object that ties together a voice port and the telephone number of a device attached to the port (also called *local dial peer*).

**PSTN** Public Switched Telephone Network. PSTN refers to the local telephone company. Sometimes called plain old telephone service (POTS).

**PVC** Permanent virtual circuit. Virtual circuit that is permanently established and is torn down in situations where certain virtual circuits must exist all the time. PVCs save bandwidth associated with circuit establishment.

## Q

**QoS** Quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

## R

**RSVP** Resource Reservation Protocol. A network protocol that enables routers to reserve the bandwidth necessary for reliable performance.

**RTCP** RTP Control Protocol. A protocol that monitors the QoS of an IPv6 RTP connection and conveys information about the on-going session.

**RTP** Real-Time Transport Protocol. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services.

## S

**SNMP** Simple Network Management Protocol. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security.

**SVC** Switched virtual circuit. Virtual circuit that is dynamically established on demand and that is torn down when transmission is complete.

## T

**Trunk** — Service that provides quasi-transparent connections between two PBXs, a PBX and a local extension, or some other combination of telephony interfaces to be permanently conferenced together by the session application and signaling passed transparently through the IP network.

## U

**UDP** — User Datagram Protocol. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols.

## V

**VIC** — Voice interface card. VICs install in a slot in the router, and provide the connection to the telephone equipment or network.

**VoIP** — Voice-over-IP, a feature that carries voice traffic, such as telephone calls and faxes, over an IP network, simultaneously with data traffic.

**VoIP dial peer** — Software object that ties together an IP address and a telephone number at a remote site reached over the IP network (also called *remote dial peer*).

**VPM** — Virtual voice-port module.

**VTSP** — Voice telephony service provider.

## A

accounting
  per VLAN **6-3**
  Quality of Service (QoS) **6-3**
acc-qos command **4-4**
addressing, in VLANs **6-4**
ADPCM CODEC **1-3**
analog signals **1-3**
answer-address command **4-5**
API **5-2**
appletalk cable-range command **7-2**
appletalk routing eigrp command **7-2**
appletalk zone command **7-2**
audience **xi**

## B

Bc **2-26**
Be **2-26**
BECN **2-25**
bridging domain **6-1**
broadcast
  control **6-3**
  domain **6-1**
  Layer 2 **6-2**
  management, in VLANs **6-5**

## C

call leg **2-9**
CELP CODEC **1-3**
central office (CO) **1-6**
CIR **2-24**
Cisco IOS software documentation **xi**
clear vlan statistics command **7-6**
CODEC
  applied **1-2**
  command **4-6**
  configuring **2-23**
  described **1-3**
codec command **4-6**
color
  See VLANs
comfort-noise command **4-7**
command conventions **xiv**
commands, debug **5-1 to 5-19**
commands, VoIP **4-1 to 4-68**
configuration
  examples **3-1**

tasks **2-2**
configuring
CODEC and VAD **2-23**
custom queuing **2-7**
dial peers **2-9**
Frame Relay for VoIP **2-24**
IP networks for real-time voice traffic **2-2**
Multilink PPP interleaving **2-4**
number expansion **2-8**
POTS dial peer **2-12**
RSVP for Voice **2-3**
RTP header compression **2-6**
voice ports **2-14**
VoIP **2-1 to 2-27**
VoIP dial peer **2-13**
weighted fair queuing **2-7**
connection command **4-8**
conventions, command **xiv**
cptone command **4-10**
custom queuing **2-7**


## D


debug cch323 h225 command **2-14**
debug cch323 rtp command **2-14**
debug commands
additional documentation **5-2**
caution **5-2**
listed **5-1**
turning off **5-2**
using in a Telnet session **5-2**
when to use **5-1**
debug vlan packet command **7-6**
debug voip ccapi error command **5-2**
debug voip ccapi inout command **5-2**
debug vpm all command **5-5**
debug vpm dsp command **5-5**
debug vpm port command **5-6**
debug vpm signal command **5-7**
debug vpm spi command **2-14, 5-8**
debug vtsp all command **5-10**
debug vtsp dsp command **5-11**
debug vtsp session command **5-16, 5-19**
delay **1-4**
description command **4-11**
destination-pattern command **4-12**
dial-control-mib command **4-13**
dial-peer configuration
optimizing **2-21**
POTS **2-12, 2-13**
table **2-12**
troubleshooting tips **2-14**
verifying **2-14**
dial peers
configuring **2-9**
described **2-9**
inbound versus outbound **2-10**
types **2-10**

dial-peer voice command **4-13**
dial-type command **4-14**
digital signal processor
  see DSP
digital signals **1-3**
DLCI **2-24**
DNS **2-26, 4-33**
documentation
  CD ROM **xi**
domain
  bridging **6-1**
  broadcast **6-1**
DSP
  debug vpm dsp command **5-5**
  defined **1-1**
  interface information **4-40**
  voice channel status **4-49**
DTMF **1-2, 4-15**

## E

E&M voice port
  configuration example **3-5**
  configuring **2-18**
  fine-tuning commands **2-20**
  signaling type **1-6**
  troubleshooting tips **2-20**
  verifying **2-19**
E.164 **1-2**
Echo **1-5**
echo-cancel coverage command **4-15**
echo-cancel enable command **4-16**
EEPROM **4-43**
encapsulation dot1q command **7-7**
examples
  Frame Relay for VoIP **2-25**
  VoIP configuration **3-1**
exit command **4-14**
expect-factor command **4-17**

## F

Fancy Queuing **2-2**
fax-rate command **4-18**
Frame Relay for VoIP
  configuring **2-24**
  example **2-25**
frame tagging, VLANs **6-3**
FXS/FXO voice ports
  configuration examples
    FXO gateway to PSTN **3-7**
    FXO gateway to PSTN (PLAR mode) **3-9**
    FXS-to-FXS connection using RSVP **3-1**
  configuring **2-15**
  fine-tuning commands **2-16**
  signaling type **1-6**
  troubleshooting tips **2-16**
  verifying **2-16**

## G

ground start signaling **1-6, 4-57**

## H

H.323 **1-1, 1-2**
hybrid switching environments **6-4**

## I

icpif command **4-19**
impedance command **4-20**
input gain command **4-21**
interface command **7-2, 7-3, 7-4**
inter-VLAN communication **6-3**
IOS software documentation **xi**
IP **1-2, 2-6**
ip precedence command **4-22**
ip rsvp bandwidth command **2-3**
ip rtp compression connections command **2-7**
ip rtp header-compression command **2-7**
ip udp checksum command **4-22**
ipx network encapsulation command **7-4**
ipx routing command **7-4**
ITU-T **1-1**

## J

jitter **1-5**

## L

LAN **6-1**
  segmentation **6-2**
    with VLANs **6-5**
Layer 2, encapsulating interfaces **6-4**
load balancing in VLANs **6-4**
loop start signaling **1-6, 4-57**
LPC CODEC **1-3**

## M

mean opinion score **1-3**
MP-MLQ CODEC **1-3**
MTU **2-24**
Multilink PPP Interleaving **2-4**
music-threshold command **4-23**

## N

NANP **1-2**
NetMeeting configuring **2-26**
network
  changes **6-3, 6-4**
  design **6-4**

- management **6-3**
  - VlanDirector **6-3**
- performance **6-4**
- scalability **6-4**
- security **6-4**
- services
  - accounting **6-3**
  - quality of service (QoS) **6-3**
  - security filtering **6-3**
- topology **6-4**

networks, switched **6-5**
non-linear command **4-24**
North American Numbering Plan **1-2**
number expansion
- command **2-8**
- configuring **2-9**
- described **2-8**
- table **2-8**

numbering scheme **1-2**
num-exp command **4-25**


## O


operation command **4-25**
organization, document **xiv**
output attenuation command **4-26**


## P


packets, VLANs **7-6**
PCM CODEC **1-3**
performance **6-3, 6-4**
PLAR connection **4-8**
port command **4-27**
POTS dial peer
- configuring **2-12**
- described **2-10**

prefix command **4-28**
PVC **2-24**


## Q


QoS
- see Quality of Service

Quality of Service
- backbone routers **2-3**
- commands
  - acc-qos **4-4**
  - req-qos **4-29**
- described **2-2**
- edge routers **2-2**
- tools
  - custom queuing **2-7**
  - listed **2-3**
  - Multilink PPP Interleaving **2-4**
  - RSVP **2-3**

RTP header compression **2-6**
weighted fair queuing **2-7**

## R

Random Early Detection **2-2**
RED
see Random Early Detection
redundancy in VLANs **6-4**
req-qos command **4-29**
resources, sharing between VLANs **6-4**
ring frequency command **4-30**
ring number command **4-31**
route
distribution **6-5**
processing **6-5**
routers, in switched VLANs **6-5**
routing between VLANs **6-4**
RSVP
applied **1-2**
configuring for voice **2-3**
enabled **2-3**
FXS-to-FXS connection example **3-1**
req-qos command **4-29**
RTCP **1-2**
RTP **1-2, 2-6**
RTP header compression **2-6**

## S

scalability, in VLANs **6-4**
security **6-4**
filtering **6-3**
VLANs **6-2**
segmentation **6-1, 6-2**
with VLANs **6-5**
session protocol command **4-32**
session target command **4-32**
session target dns command **4-33**
session target loopback command **4-33**
show call active voice command **4-34**
show call history voice command **4-37**
show dial-peer voice command **4-45**
show dialplan incall number command **4-47**
show dialplan number command **4-48**
show num-exp command **4-48**
show vlans command **7-6, 7-7**
show voice port command **4-50**
shutdown (dial peer) command **4-55**
shutdown (voice port) command **4-56**
signal command **4-56**
signaling types
E&M **1-6, 2-15**
FXS/FXO **1-6, 2-15**
SNMP
event **4-4**
status change **4-61**

trap message, generating **2-23**
trap operation, enabling **4-59**
snmp enable peer-trap poor-qov command **4-58**
snmp-server enable traps command **4-59**
snmp-server host command **4-59**
snmp trap link-status command **4-60**

## T

timeouts initial command **4-61**
timeouts interdigit command **4-62**
timing command **4-63**
traffic
broadcast **6-3**
controlling patterns **6-4**
multicast **6-3**
traffic shaping in Frame Relay **2-25**
translation, in VLANs **6-4**
troubleshooting
dial-peer configuration **2-14**
E&M configuration **2-20**
FXS/FXO configuration **2-16**
trunk connection **4-8**
type command **4-65**

## U

UDP **1-2, 2-6**

## V

VAD
configuring **2-24**
described **2-23**
effect on comfort-noise command **4-8**
effect on music-threshold command **4-23**
vad command **4-67**
VFC modem **5-11**
VIC
described **2-14**
slot information **4-43**
virtual LANs
See VLANs
virtual voice-port module **5-5**
VLANs
addressing **6-4**
broadcast domain **6-1**
colors **6-3**
communication between **6-3**
debug vlan packet command **7-6**
description **6-1**
designing switched VLANs **6-4**
frame tagging **6-3**
hybrid switching environments **6-4**

identifier **6-3**
isolation between **6-3**
LAN segmentation **6-5**
load balancing **6-4**
monitoring **7-7**
network
changes **6-4**
design **6-4**
management **6-3**
performance **6-3**
performance **6-4**
redundancy in **6-4**
routers in **6-5**
routing between **6-4**
scalability **6-2, 6-4**
security **6-2, 6-4**
segmenting LANs with **6-1, 6-2**
sharing resources between **6-4**
translation **6-4**
VlanDirector **6-3**
voice activity detection
see VAD
voice interface card
see VIC
Voice over IP
commands **4-1 to 4-68**
configuration examples **3-1 to 3-10**
configuring **2-1 to 2-27**
debug commands **5-1 to 5-19**
Frame Relay, configuring for **2-24**
Microsoft NetMeeting, configuring for **2-26**
voice-port command **4-67**
voice ports
commands **4-3**
E&M
configuring **2-18**
described **2-15**
fine-tuning commands **2-20**
troubleshooting tips **2-20**
verifying **2-19**
FXS/FXO
configuring **2-15**
described **2-15**
fine-tuning commands **2-16**
troubleshooting tips **2-16**
verifying **2-16**
VoIP
see Voice over IP
VoIP dial peer
configuring **2-13**
described **2-10**
VPM **5-5**

## W

weighted fair queuing **2-7**
Weighted Random Early Detection **2-2**

WRED

see Weighted Random Early Detection